# OpenVMS Cluster Systems

Order Number: AA–PV5WF–TK

**June 2002**

This manual describes procedures and guidelines for configuring and managing OpenVMS Cluster systems. Except where noted, the procedures and guidelines apply equally to VAX and Alpha computers. This manual also includes information for providing high availability, building-block growth, and unified system management across coupled systems.

**Revision/Update Information:** This manual supersedes *OpenVMS Cluster Systems*, OpenVMS Alpha Version 7.3 and OpenVMS VAX Version 7.3.

**Software Version:** OpenVMS Alpha Version 7.3–1
OpenVMS VAX Version 7.3

**Compaq Computer Corporation**
**Houston, Texas**

ZK4477

The Compaq *OpenVMS* documentation set is available on CD-ROM.

This document was prepared using DECdocument, Version 3.3-1b.

# Contents

## 3 OpenVMS Cluster Interconnect Configurations

## 4 The OpenVMS Cluster Operating Environment



## 5 Preparing a Shared Environment

## 6 Cluster Storage Devices

## 7 Setting Up and Managing Cluster Queues



## 8 Configuring an OpenVMS Cluster System

## 9 Building Large OpenVMS Cluster Systems

## 10 Maintaining an OpenVMS Cluster System



## A Cluster System Parameters

## B Building Common Files



## C Cluster Troubleshooting

## D Sample Programs for LAN Control



## E Subroutines for LAN Control



## F Troubleshooting the NISCA Protocol

## G NISCA Transport Protocol Channel Selection and Congestion Control



## Index

## Examples

## Figures

## Tables

# Preface

## Introduction

*OpenVMS Cluster Systems* describes system management for OpenVMS Cluster systems. Although the OpenVMS Cluster software for VAX and Alpha computers is separately purchased, licensed, and installed, the difference between the two architectures lies mainly in the hardware used. Essentially, system management for VAX and Alpha computers in an OpenVMS Cluster is identical. Exceptions are pointed out.

## Who Should Use This Manual

This document is intended for anyone responsible for setting up and managing OpenVMS Cluster systems. To use the document as a guide to cluster management, you must have a thorough understanding of system management concepts and procedures, as described in the *OpenVMS System Manager's Manual*.

## How This Manual Is Organized

*OpenVMS Cluster Systems* contains ten chapters and seven appendixes.

Chapter 1 introduces OpenVMS Cluster systems.

Chapter 2 presents the software concepts integral to maintaining OpenVMS Cluster membership and integrity.

Chapter 3 describes various OpenVMS Cluster configurations and the ways they are interconnected.

Chapter 4 explains how to set up an OpenVMS Cluster system and coordinate system files.

Chapter 5 explains how to set up an environment in which resources can be shared across nodes in the OpenVMS Cluster system.

Chapter 6 discusses disk and tape management concepts and procedures and how to use Volume Shadowing for OpenVMS to prevent data unavailability.

Chapter 7 discusses queue management concepts and procedures.

Chapter 8 explains how to build an OpenVMS Cluster system once the necessary preparations are made, and how to reconfigure and maintain the cluster.

Chapter 9 provides guidelines for configuring and building large OpenVMS Cluster systems, booting satellite nodes, and cross-architecture booting.

Chapter 10 describes ongoing OpenVMS Cluster system maintenance.

Appendix A lists and defines OpenVMS Cluster system parameters.

Appendix B provides guidelines for building a cluster common user authorization file.

Appendix C provides troubleshooting information.

Appendix D presents three sample programs for LAN control and explains how to use the Local Area OpenVMS Cluster Network Failure Analysis Program.

Appendix E describes the subroutine package used with local area OpenVMS Cluster sample programs.

Appendix F provides techniques for troubleshooting network problems related to the NISCA transport protocol.

Appendix G describes how the interactions of workload distribution and network topology affect OpenVMS Cluster system performance, and discusses transmit channel selection by PEDRIVER.

## Associated Documents

This document is not a one-volume reference manual. The utilities and commands are described in detail in the *OpenVMS System Manager's Manual*, the *OpenVMS System Management Utilities Reference Manual*, and the *OpenVMS DCL Dictionary*.

For additional information on the topics covered in this manual, refer to the following documents:

- *Guidelines for OpenVMS Cluster Configurations*
- *OpenVMS Alpha Partitioning and Galaxy Guide*
- *Guide to OpenVMS File Applications*
- *OpenVMS Guide to System Security*
- *OpenVMS Alpha System Dump Analyzer Utility Manual*
- *VMS System Dump Analyzer Utility Manual*
- *OpenVMS I/O User's Reference Manual*
- *OpenVMS License Management Utility Manual*
- *OpenVMS System Management Utilities Reference Manual*
- *OpenVMS System Manager's Manual*
- *A Comparison of System Management on OpenVMS AXP and OpenVMS VAX*[1]
- *OpenVMS System Services Reference Manual*
- *Volume Shadowing for OpenVMS*
- OpenVMS Cluster Software *Software Product Description* (SPD 29.78.*xx*)
- *DECnet for OpenVMS Network Management Utilities*
- *DECnet for OpenVMS Networking Manual*
- The DECnet–Plus (formerly known as DECnet/OSI) documentation set
- The TCP/IP Services for OpenVMS documentation set

For additional information about Compaq *OpenVMS* products and services, access the Compaq website at the following location:

`http://www.openvms.compaq.com/`

[1] This manual has been archived but is available on the OpenVMS Documentation CD–ROM.

## Reader's Comments

Compaq welcomes your comments on this manual. Please send comments to either of the following addresses:

| | |
|---|---|
| Internet | **openvmsdoc@compaq.com** |
| Mail | Compaq Computer Corporation<br>OSSG Documentation Group, ZKO3-4/U08<br>110 Spit Brook Rd.<br>Nashua, NH 03062-2698 |

## How To Order Additional Documentation

Visit the following World Wide Web address for information about how to order additional documentation:

`http://www.openvms.compaq.com/`

### Conventions

The following conventions are used in this manual:

| | |
|---|---|
| Return | In examples, a key name enclosed in a box indicates that you press a key on the keyboard. (In text, a key name is not enclosed in a box.)<br>In the HTML version of this document, this convention appears as brackets, rather than a box. |
| . . . | A horizontal ellipsis in examples indicates one of the following possibilities:<br>• Additional optional arguments in a statement have been omitted.<br>• The preceding item or items can be repeated one or more times.<br>• Additional parameters, values, or other information can be entered. |
| .<br>.<br>. | A vertical ellipsis indicates the omission of items from a code example or command format; the items are omitted because they are not important to the topic being discussed. |
| ( ) | In command format descriptions, parentheses indicate that you must enclose choices in parentheses if you specify more than one. |
| [ ] | In command format descriptions, brackets indicate optional choices. You can choose one or more items or no items. Do not type the brackets on the command line. However, you must include the brackets in the syntax for OpenVMS directory specifications and for a substring specification in an assignment statement. |
| \| | In command format descriptions, vertical bars separate choices within brackets or braces. Within brackets, the choices are optional; within braces, at least one choice is required. Do not type the vertical bars on the command line. |
| { } | In command format descriptions, braces indicate required choices; you must choose at least one of the items listed. Do not type the braces on the command line. |

| | |
|---|---|
| **bold text** | This typeface represents the introduction of a new term. It also represents the name of an argument, an attribute, or a reason. |
| *italic text* | Italic text indicates important information, complete titles of manuals, or variables. Variables include information that varies in system output (Internal error *number*), in command lines (/PRODUCER=*name*), and in command parameters in text (where *dd* represents the predefined code for the device type). |
| UPPERCASE TEXT | Uppercase text indicates a command, the name of a routine, the name of a file, or the abbreviation for a system privilege. |
| `Monospace text` | Monospace type indicates code examples and interactive screen displays.<br>In the C programming language, monospace type in text identifies the following elements: keywords, the names of independently compiled external functions and files, syntax summaries, and references to variables or identifiers introduced in an example. |
| - | A hyphen at the end of a command format description, command line, or code line indicates that the command or statement continues on the following line. |
| numbers | All numbers in text are assumed to be decimal unless otherwise noted. Nondecimal radixes—binary, octal, or hexadecimal—are explicitly indicated. |

# 1

# Introduction to OpenVMS Cluster System Management

"Cluster" technology was pioneered by Digital Equipment Corporation in 1983 with the VAXcluster system. The VAXcluster system was built using multiple standard VAX computing systems and the VMS operating system. The initial VAXcluster system offered the power and manageability of a centralized system and the flexibility of many physically distributed computing systems.

Through the years, the technology has evolved into OpenVMS Cluster systems, which support both the OpenVMS Alpha and the OpenVMS VAX operating systems and hardware, as well as a multitude of additional features and options. When Compaq Computer Corporation acquired Digital Equipment Corporation in 1998, it acquired the most advanced cluster technology available. Compaq continues to enhance and expand OpenVMS Cluster capabilities.

## 1.1 Overview

An **OpenVMS Cluster system** is a highly integrated organization of OpenVMS software, Alpha and VAX computers, and storage devices that operate as a single system. The OpenVMS Cluster acts as a single virtual system, even though it is made up of many distributed systems. As members of an OpenVMS Cluster system, Alpha and VAX computers can share processing resources, data storage, and queues under a single security and management domain, yet they can boot or shut down independently.

The distance between the computers in an OpenVMS Cluster system depends on the interconnects that you use. The computers can be located in one computer lab, on two floors of a building, between buildings on a campus, or on two different sites hundreds of miles apart.

An OpenVMS Cluster system with computers located on two different sites is known as a multiple-site OpenVMS Cluster system. A multiple-site OpenVMS Cluster forms the basis of a disaster tolerant OpenVMS Cluster system. For more information about multiple site clusters, refer to *Guidelines for OpenVMS Cluster Configurations*.

Disaster Tolerant Cluster Services for OpenVMS is a Compaq Services system management and software package for configuring and managing OpenVMS disaster tolerant clusters. For more information about Disaster Tolerant Cluster Services for OpenVMS, contact your Compaq Services representative.

You can also visit:

`http://www.compaq.co.uk/globalservices/continuity/dis_clus.stm`

### 1.1.1 Uses

OpenVMS Cluster systems are an ideal environment for developing high-availability applications, such as transaction processing systems, servers for network client/server applications, and data-sharing applications.

### 1.1.2 Benefits

Computers in an OpenVMS Cluster system interact to form a cooperative, distributed operating system and derive a number of benefits, as shown in the following table.

| Benefit | Description |
|---|---|
| Resource sharing | OpenVMS Cluster software automatically synchronizes and load balances batch and print queues, storage devices, and other resources among all cluster members. |
| Flexibility | Application programmers do not have to change their application code, and users do not have to know anything about the OpenVMS Cluster environment to take advantage of common resources. |
| High availability | System designers can configure redundant hardware components to create highly available systems that eliminate or withstand single points of failure. |
| Nonstop processing | The OpenVMS operating system, which runs on each node in an OpenVMS Cluster, facilitates dynamic adjustments to changes in the configuration. |
| Scalability | Organizations can dynamically expand computing and storage resources as business needs grow or change without shutting down the system or applications running on the system. |
| Performance | An OpenVMS Cluster system can provide high performance. |
| Management | Rather than repeating the same system management operation on multiple OpenVMS systems, management tasks can be performed concurrently for one or more nodes. |
| Security | Computers in an OpenVMS Cluster share a single security database that can be accessed by all nodes in a cluster. |
| Load balancing | Distributes work across cluster members based on the current load of each member. |

## 1.2 Hardware Components

OpenVMS Cluster system configurations consist of hardware components from the following general groups:

- Computers
- Interconnects
- Storage devices

**References:** Detailed OpenVMS Cluster configuration guidelines can be found in the OpenVMS Cluster Software *Software Product Description* (SPD) and in *Guidelines for OpenVMS Cluster Configurations*.

### 1.2.1 Computers

Up to 96 computers, ranging from desktop to mainframe systems, can be **members** of an OpenVMS Cluster system. Active members that run the OpenVMS Alpha or OpenVMS VAX operating system and participate fully in OpenVMS Cluster negotiations can include:

- Alpha computers or workstations
- VAX computers or workstations or MicroVAX computers

### 1.2.2 Physical Interconnects

An **interconnect** is a physical path that connects computers to other computers and to storage subsystems. OpenVMS Cluster systems support a variety of interconnects (also referred to as buses) so that members can communicate using the most appropriate and effective method possible:

- LANs
  - Ethernet (10/100, Gigabit)
  - Asynchronous transfer mode (ATM)
  - Fiber Distributed Data Interface (FDDI)
- CI
- Digital Storage Systems Interconnect (DSSI)
- MEMORY CHANNEL (node to node only)
- Small Computer Systems Interface (SCSI) (storage only)
- Fibre Channel (FC) (storage only)

### 1.2.3 OpenVMS Galaxy SMCI

In addition to the physical interconnects listed in Section 1.2.2, another type of interconnect, a shared memory CI (SMCI) for OpenVMS Galaxy instances, is available. SMCI supports cluster communications between Galaxy instances.

For more information about SMCI and Galaxy configurations, see the *OpenVMS Alpha Partitioning and Galaxy Guide*.

### 1.2.4 Storage Devices

A **shared** storage device is a disk or tape that is accessed by multiple computers in the cluster. Nodes access remote disks and tapes by means of the MSCP and TMSCP server software (described in Section 1.3.1).

Systems within an OpenVMS Cluster support a wide range of storage devices:

- Disks and disk drives, including:
  - Digital Storage Architecture (DSA) disks
  - RF series integrated storage elements (ISEs)
  - Small Computer Systems Interface (SCSI) devices
  - Solid state disks
- Tapes and tape drives

- Controllers and I/O servers, including the following:

| Controller | Interconnect |
|---|---|
| HSC | CI |
| HSJ | CI |
| HSD | DSSI |
| HSZ | SCSI |
| HSG | FC |

In addition, the K.scsi HSC controller allows the connection of the StorageWorks arrays with SCSI devices on the HSC storage subsystems.

**Note:** HSC, HSJ, HSD, and HSZ controllers support many combinations of SDIs (standard disk interfaces) and STIs (standard tape interfaces) that connect disks and tapes.

## 1.3 Software Components

The OpenVMS operating system, which runs on each node in the OpenVMS Cluster, includes several software components that facilitate resource sharing and dynamic adjustments to changes in the underlying hardware configuration.

If one computer becomes unavailable, the OpenVMS Cluster system continues operating because OpenVMS is still running on the remaining computers.

### 1.3.1 OpenVMS Cluster Software Functions

The following table describes the software components and their main function.

| Component | Facilitates | Function |
|---|---|---|
| Connection manager | Member integrity | Coordinates participation of computers in the cluster and maintains cluster integrity when computers join or leave the cluster. |
| Distributed lock manager | Resource synchronization | Synchronizes operations of the distributed file system, job controller, device allocation, and other cluster facilities. If an OpenVMS Cluster computer shuts down, all locks that it holds are released so processing can continue on the remaining computers. |
| Distributed file system | Resource sharing | Allows all computers to share access to mass storage and file records, regardless of the type of storage device (DSA, RF, SCSI, and solid state subsystem) or its location. |
| Distributed job controller | Queuing | Makes generic and execution queues available across the cluster. |
| MSCP server | Disk serving | Implements the proprietary mass storage control protocol in order to make disks available to all nodes that do not have direct access to those disks. |
| TMSCP server | Tape serving | Implements the proprietary tape mass storage control protocol in order to make tape drives available to all nodes that do not have direct access to those tape drives. |

## 1.4 Communications

The System Communications Architecture (SCA) defines the communications mechanisms that allow nodes in an OpenVMS Cluster system to cooperate. It governs the sharing of data between resources at the nodes and binds together System Applications (SYSAPs) that run on different Alpha and VAX computers.

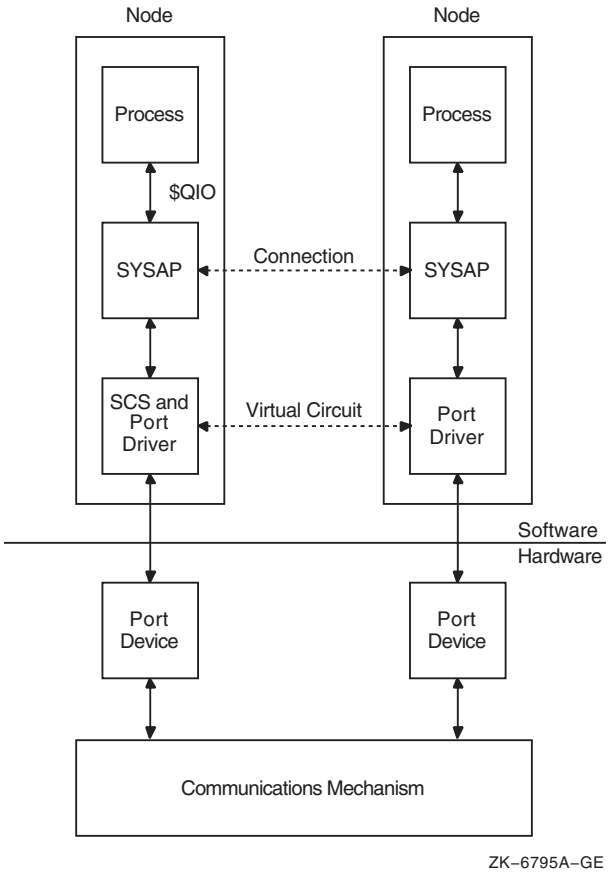
SCA consists of the following hierarchy of components:

| Communications Software | Function |
|---|---|
| System applications (SYSAPs) | Consists of clusterwide applications (for example, disk and tape class drivers, connection manager, and MSCP server) that use SCS software for interprocessor communication. |
| System Communications Services (SCS) | Provides basic connection management and communication services, implemented as a logical path, between system applications (SYSAPs) on nodes in an OpenVMS Cluster system. |
| Port drivers | Control the communication paths between local and remote ports. |
| Physical interconnects | Consists of ports or adapters for CI, DSSI, Ethernet (10/100 and Gigabit), ATM, FDDI, and MEMORY CHANNEL interconnects. |

### 1.4.1 System Communications

Figure 1–1 shows the interrelationships between the OpenVMS Cluster components.

**Figure 1–1 OpenVMS Cluster System Communications**




In Figure 1–1, processes in different nodes exchange information with each other:

- Processes can call the $QIO system service and other system services directly from a program or indirectly using other mechanisms such as OpenVMS Record Management Services (RMS). The $QIO system service initiates all I/O requests.
- A SYSAP on one OpenVMS Cluster node must communicate with a SYSAP on another node. For example, a connection manager on one node must communicate with the connection manager on another node, or a disk class driver on one node must communicate with the MSCP server on another node.
- The following SYSAPs use SCS for cluster communication:
  - Disk and tape class drivers
  - MSCP server
  - TMSCP server
  - DECnet class driver
  - Connection manager
- SCS routines provide services to format and transfer SYSAP messages to a port driver for delivery over a specific interconnect.

- Communications go through the port drivers to OpenVMS Cluster computers and storage controllers. The port driver manages a logical path, called a **virtual circuit**, between each pair of ports in an OpenVMS Cluster system.

### 1.4.2 Application Communications

Applications running on OpenVMS Cluster systems use DECnet or TCP/IP (transmission control protocol and internet protocol) for application communication. The DECnet and TCP/IP communication services allow processes to locate or start remote servers and then exchange messages.

Note that generic references to DECnet in this document mean either DECnet for OpenVMS or DECnet–Plus (formerly known as DECnet/OSI) software.

### 1.4.3 Cluster Alias

A DECnet feature known as a **cluster alias** provides a collective name for the nodes in an OpenVMS Cluster system. Application software can connect to a node in the OpenVMS Cluster using the cluster alias name rather than a specific node name. This frees the application from keeping track of individual nodes in the OpenVMS Cluster system and results in design simplification, configuration flexibility, and application availability.

## 1.5 System Management

The OpenVMS Cluster system manager must manage multiple users and resources for maximum productivity and efficiency while maintaining the necessary security.

### 1.5.1 Ease of Management

An OpenVMS Cluster system is easily managed because the multiple members, hardware, and software are designed to cooperate as a single system:

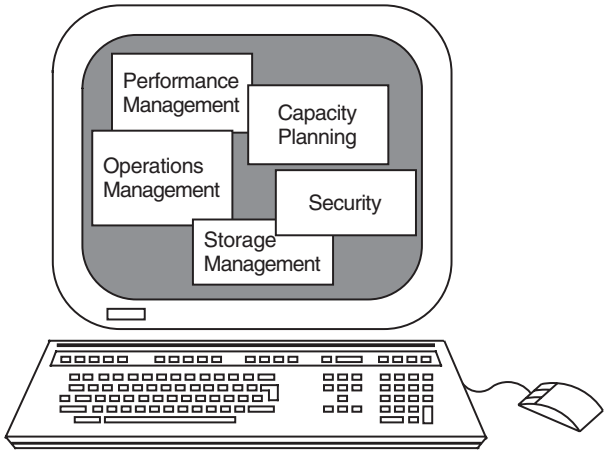
- Smaller configurations usually include only one system disk (or two for an OpenVMS Cluster configuration with both OpenVMS VAX and OpenVMS Alpha operating systems), regardless of the number or location of computers in the configuration.
- Software needs to be installed only once for each operating system (VAX and Alpha), and is accessible by every user and node of the OpenVMS Cluster.
- Users need to be added once to have access to the resources of the entire OpenVMS Cluster.
- Several system management utilities and commands facilitate cluster management.

Figure 1–2 illustrates centralized system management.

### 1.5.2 Tools and Utilities from Compaq

The OpenVMS operating system supports a number of utilities and tools to assist you with the management of the distributed resources in OpenVMS Cluster configurations. Proper management is essential to ensure the availability and performance of OpenVMS Cluster configurations.

**Figure 1–2 Single-Point OpenVMS Cluster System Management**




OpenVMS and its partners offer a wide selection of tools to meet diverse system management needs. Table 1–1 describes the Compaq products available for cluster management and indicates whether each is supplied with the operating system or is an optional product. Table 1–2 describes some of the most important utilities and tools produced by OpenVMS partners for managing an OpenVMS Cluster configuration.

**Table 1–1 System Management Tools**

| Tool | Supplied or Optional | Function |
|---|---|---|
| **Accounting** | | |
| VMS Accounting | Supplied | Tracks how resources are being used. |
| **Configuration and capacity planning** | | |
| LMF (License Management Facility) | Supplied | Helps the system manager to determine which software products are licensed and installed on a standalone system and on each of the computers in an OpenVMS Cluster system. |
| Graphical Configuration Manager (GCM) | Supplied | A portable client/server application that gives you a way to view and control the configuration of partitioned AlphaServer systems running OpenVMS. |
| Galaxy Configuration Utility (GCU) | Supplied | A DECwindows Motif application that allows system managers to configure and manage an OpenVMS Galaxy system from a single workstation window. |
| SYSGEN (System Generation) utility | Supplied | Allows you to tailor your system for a specific hardware and software configuration. Use SYSGEN to modify system parameters, load device drivers, and create additional page and swap files. |
| CLUSTER_CONFIG.COM | Supplied | Automates the configuration or reconfiguration of an OpenVMS Cluster system and assumes the use of DECnet. |
| CLUSTER_CONFIG_LAN.COM | Supplied | Automates configuration or reconfiguration of an OpenVMS Cluster system without the use of DECnet. |
| Compaq Management Agents for OpenVMS | Supplied | Consists of a web server for system management with management agents that allow you to look at devices on your OpenVMS systems. |
| Compaq *Insight Manager XE* | Supplied with every Compaq NT server | Centralizes system management in one system to reduce cost, improve operational efficiency and effectiveness, and minimize system down time. You can use Compaq Insight Manager XE on an NT server to monitor every system in an OpenVMS Cluster system. In a configuration of heterogeneous Compaq systems, you can use Compaq Insight Manager XE on an NT server to monitor all systems. |
| **Event and fault tolerance** | | |
| OPCOM message routing | Supplied | Provides event notification. |
| **Operations management** | | |
| Clusterwide process services | Supplied | Allows OpenVMS system management commands, such as SHOW USERS, SHOW SYSTEM, and STOP/ID=, to operate clusterwide. |
| Availability Manager | Supplied | From either an OpenVMS Alpha or a Windows node, enables you to monitor one or more OpenVMS nodes on an extended local area network (LAN). Availability Manager collects system and process data from multiple OpenVMS nodes simultaneously, then analyzes the data, and displays the output using a native Java GUI. |

(continued on next page)

**Table 1–1 (Cont.) System Management Tools**

| Tool | Supplied or Optional | Function |
|---|---|---|
| **Operations management** | | |
| DECamds | Supplied | Collects and analyzes data from multiple nodes simultaneously, directing all output to a centralized DECwindows display. The analysis detects resource availability problems and suggests corrective actions. |
| SCACP (Systems Communications Architecture Control Program) | Supplied | Enables you to monitor and manage switched LAN paths. |
| DFS (Distributed File Service) | Optional | Allows disks to be served across a LAN or WAN. |
| DNS (Distributed Name Service) | Optional | Configures certain network nodes as name servers that associate objects with network names. |
| LATCP (Local Area Transport Control Program) | Supplied | Provides the function to control and obtain information from the LAT port driver. |
| LANCP (LAN Control Program) | Supplied | Allows the system manager to configure and control the LAN software on OpenVMS systems. |
| NCP (Network Control Protocol) utility | Optional | Allows the system manager to supply and access information about the DECnet for OpenVMS (Phase IV) network from a configuration database. |
| NCL (Network Control Language) utility | Optional | Allows the system manager to supply and access information about the DECnet–Plus network from a configuration database. |
| OpenVMS Management Station | Supplied | Enables system managers to set up and manage accounts and print queues across multiple OpenVMS Cluster systems and OpenVMS nodes. OpenVMS Management Station is a Microsoft Windows and Windows NT based management tool. |
| POLYCENTER Software Installation Utility (PCSI) | Supplied | Provides rapid installations of software products. |
| Queue Manager | Supplied | Uses OpenVMS Cluster generic and execution queues to feed node-specific queues across the cluster. |
| Show Cluster utility | Supplied | Monitors activity and performance in an OpenVMS Cluster configuration, then collects and sends information about that activity to a terminal or other output device. |
| SDA (System Dump Analyzer) | Supplied | Allows you to inspect the contents of memory as saved in the dump taken at crash time or as it exists in a running system. You can use SDA interactively or in batch mode. |
| SYSMAN (System Management utility) | Supplied | Enables device and processor control commands to take effect across an OpenVMS Cluster. |
| VMSINSTAL | Supplied | Provides software installations. |

(continued on next page)

**Table 1–1 (Cont.) System Management Tools**

| Tool | Supplied or Optional | Function |
|---|---|---|
| **Performance** | | |
| AUTOGEN utility | Supplied | Optimizes system parameter settings based on usage. |
| Monitor utility | Supplied | Provides basic performance data. |
| **Security** | | |
| Authorize utility | Supplied | Modifies user account profiles. |
| SET ACL command | Supplied | Sets complex protection on many system objects. |
| SET AUDIT command | Supplied | Facilitates tracking of sensitive system objects. |
| **Storage management** | | |
| Backup utility | Supplied | Allows OpenVMS Cluster system managers to create backup copies of files and directories from storage media and then restore them. This utility can be used on one node to back up data stored on disks throughout the OpenVMS Cluster system. |
| Mount utility | Supplied | Enables a disk or tape volume for processing by one computer, a subset of OpenVMS Cluster computers, or all OpenVMS Cluster computers. |
| Volume Shadowing for OpenVMS | Optional | Replicates disk data across multiple disks to help OpenVMS Cluster systems survive disk failures. |

### 1.5.3 System Management Tools from OpenVMS Partners

OpenVMS Partners offer a wide selection of tools to meet diverse system management needs, as shown in Table 1–2. The types of tools are described in the following list:

- Schedule managers

  Enable specific actions to be triggered at determined times, including repetitive and periodic activities, such as nightly backups.

- Event managers

  Monitor a system and report occurrences and events that may require an action or that may indicate a critical or alarming situation, such as low memory or an attempted security breakin.

- Console managers

  Enable a remote connection to and emulation of a system console so that system messages can be displayed and commands can be issued.

- Performance managers

  Monitor system performance by collecting and analyzing data to allow proper tailoring and configuration of system resources. Performance managers may also collect historical data for capacity planning.

**Table 1–2 System Management Products from OpenVMS Partners**

| Business Partner | Product | Type or Function |
|---|---|---|
| BMC | Perform and Predict | Performance and capacity manager |
| | Patrol for OpenVMS | Event manager |
| | ControlM | Console manager |
| Computer Associates | Advise*IT* | Performance manager |
| | Command*IT* | Console manager |
| | Schedule*IT* | Schedule manager |
| | Watch*IT* | Event manager |
| | Unicenter TNG | Package of various products |
| Fortel | ViewPoint | Performance manager |
| Global Maintech | VCC | Console manager |
| Heroix | RoboMon | Event manager |
| | RoboCentral | Console manager |
| ISE | Schedule | Schedule manager |
| Ki NETWORKS | CLIM | Console manager |
| ORSYP | Dollar Universe | Schedule manager |
| RAXCO | Perfect Cache | Storage performance |
| | Perfect Disk | Storage management |
| TECsys Development Inc. | Console Works | Console manager |

For current information about OpenVMS Partners and the tools they provide, visit the following web site:
`http://www.openvms.compaq.com/openvms/system_management.html`

### 1.5.4 Other Configuration Aids

In addition to these utilities and partner products, several commands are available that allow the system manager to set parameters on HSC, HSJ, HSD, HSZ, HSG, and RF subsystems to help configure the system. See the appropriate hardware documentation for more information.

# 2
# OpenVMS Cluster Concepts

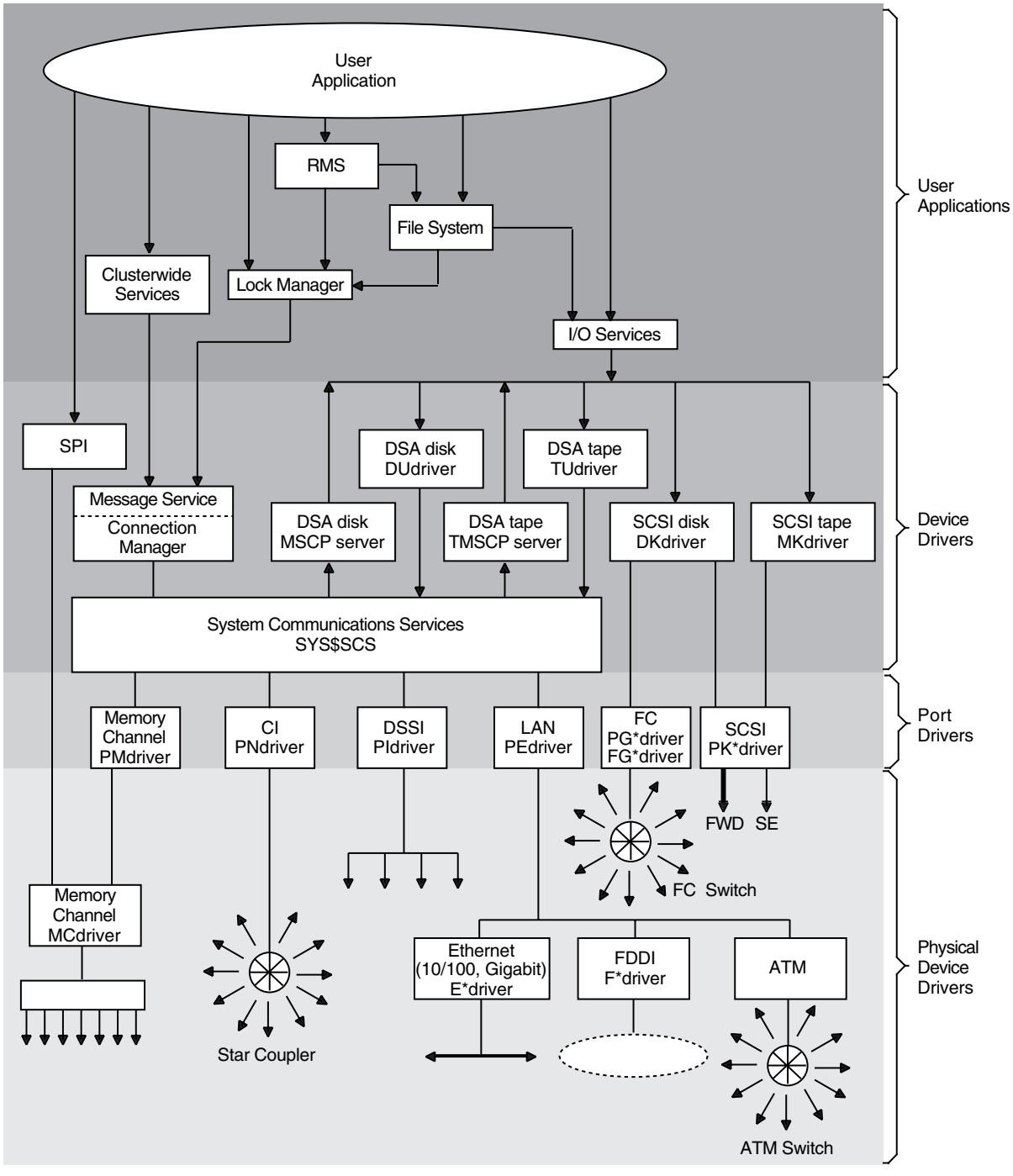
To help you understand the design and implementation of an OpenVMS Cluster system, this chapter describes its basic architecture.

## 2.1 OpenVMS Cluster System Architecture

Figure 2–1 illustrates the protocol layers within the OpenVMS Cluster system architecture, ranging from the communications mechanisms at the base of the figure to the users of the system at the top of the figure. These protocol layers include:

- Ports
- System Communications Services (SCS)
- System Applications (SYSAPs)
- Other layered components

**Figure 2–1 OpenVMS Cluster System Architecture**




## 2.1.1 Port Layer

This lowest level of the architecture provides connections, in the form of communication ports and physical paths, between devices. The port layer can contain any of the following interconnects:

- LANs
  - ATM
  - Ethernet (10/100 and Gigabit Ethernet)
  - FDDI
- CI
- DSSI
- MEMORY CHANNEL
- SCSI
- Fibre Channel

Each interconnect is accessed by a port (also referred to as an adapter) that connects to the processor node. For example, the Fibre Channel interconnect is accessed by way of a Fibre Channel port.

### 2.1.2 SCS Layer

The SCS layer provides basic connection management and communications services in the form of datagrams, messages, and block transfers over each logical path. Table 2–1 describes these services.

**Table 2–1 Communications Services**

| Service | Delivery Guarantees | Usage |
|---|---|---|
| **Datagrams** | | |
| Information units of less than one packet | Delivery of datagrams is not guaranteed. Datagrams can be lost, duplicated, or delivered out of order. | Status and information messages whose loss is not critical<br>Applications that have their own reliability protocols (such as DECnet) |
| **Messages** | | |
| Information units of less than one packet | Messages are guaranteed to be delivered and to arrive in order. Virtual circuit sequence numbers are used on the individual packets. | Disk read and write requests |
| **Block data transfers** | | |
| Any contiguous data in a process virtual address space. There is no size limit except that imposed by the physical memory constraints of the host system. | Delivery of block data is guaranteed. The sending and receiving ports and the port emulators cooperate in breaking the transfer into data packets and ensuring that all packets are correctly transmitted, received, and placed in the appropriate destination buffer. Block data transfers differ from messages in the size of the transfer. | Disk subsystems and disk servers to move data associated with disk read and write requests |

The SCS layer is implemented as a combination of hardware and software, or software only, depending upon the type of port. SCS manages connections in an OpenVMS Cluster and multiplexes messages between system applications over a common transport called a **virtual circuit**. A virtual circuit exists between each pair of SCS ports and a set of SCS connections that are multiplexed on that virtual circuit.

### 2.1.3 System Applications (SYSAPs) Layer

The next higher layer in the OpenVMS Cluster architecture consists of the SYSAPs layer. This layer consists of multiple system applications that provide, for example, access to disks and tapes and cluster membership control. SYSAPs can include:

- Connection manager
- MSCP server
- TMSCP server
- Disk and tape class drivers

These components are described in detail later in this chapter.

### 2.1.4 Other Layered Components

A wide range of OpenVMS components layer on top of the OpenVMS Cluster system architecture, including:

- Volume Shadowing for OpenVMS
- Distributed lock manager
- Process control services
- Distributed file system
- Record Management Services (RMS)
- Distributed job controller

These components, except for volume shadowing, are described in detail later in this chapter. Volume Shadowing for OpenVMS is described in Section 6.6.

## 2.2 OpenVMS Cluster Software Functions

The OpenVMS Cluster software components that implement OpenVMS Cluster communication and resource-sharing functions always run on every computer in the OpenVMS Cluster. If one computer fails, the OpenVMS Cluster system continues operating, because the components still run on the remaining computers.

### 2.2.1 Functions

The following table summarizes the OpenVMS Cluster communication and resource-sharing functions and the components that perform them.

| Function | Performed By |
|---|---|
| Ensure that OpenVMS Cluster computers communicate with one another to enforce the rules of cluster membership | Connection manager |
| Synchronize functions performed by other OpenVMS Cluster components, OpenVMS products, and other software components | Distributed lock manager |
| Share disks and files | Distributed file system |
| Make disks available to nodes that do not have direct access | MSCP server |

| Function | Performed By |
|---|---|
| Make tapes available to nodes that do not have direct access | TMSCP server |
| Make queues available | Distributed job controller |

## 2.3 Ensuring the Integrity of Cluster Membership

The connection manager ensures that computers in an OpenVMS Cluster system communicate with one another to enforce the rules of cluster membership.

Computers in an OpenVMS Cluster system share various data and system resources, such as access to disks and files. To achieve the coordination that is necessary to maintain resource integrity, the computers must maintain a clear record of cluster membership.

### 2.3.1 Connection Manager

The connection manager creates an OpenVMS Cluster when the first computer is booted and reconfigures the cluster when computers join or leave it during cluster **state transitions**. The overall responsibilities of the connection manager are to:

- Prevent partitioning (see Section 2.3.2).
- Track which nodes in the OpenVMS Cluster system are active and which are not.
- Deliver messages to remote nodes.
- Remove nodes.
- Provide a highly available message service in which other software components, such as the distributed lock manager, can synchronize access to shared resources.

### 2.3.2 Cluster Partitioning

A primary purpose of the connection manager is to prevent **cluster partitioning**, a condition in which nodes in an existing OpenVMS Cluster configuration divide into two or more independent clusters.

Cluster partitioning can result in data file corruption because the distributed lock manager cannot coordinate access to shared resources for multiple OpenVMS Cluster systems. The connection manager prevents cluster partitioning using a quorum algorithm.

### 2.3.3 Quorum Algorithm

The quorum algorithm is a mathematical method for determining if a majority of OpenVMS Cluster members exist so resources can be shared across an OpenVMS Cluster system. **Quorum** is the number of votes that must be present for the cluster to function. Quorum is a dynamic value calculated by the connection manager to prevent cluster partitioning. The connection manager allows processing to occur only if a majority of the OpenVMS Cluster members are functioning.

### 2.3.4 System Parameters

Two system parameters, VOTES and EXPECTED_VOTES, are key to the computations performed by the quorum algorithm. The following table describes these parameters.

| Parameter | Description |
|---|---|
| VOTES | Specifies a fixed number of votes that a computer contributes toward quorum. The system manager can set the VOTES parameters on each computer or allow the operating system to set it to the following default values:<br>• For satellite nodes, the default value is 0.<br>• For all other computers, the default value is 1.<br>Each Alpha or VAX computer with a nonzero value for the VOTES system parameter is considered a **voting member**. |
| EXPECTED_ VOTES | Specifies the sum of all VOTES held by OpenVMS Cluster members. The *initial* value is used to derive an estimate of the *correct* quorum value for the cluster. The system manager must set this parameter on each active Alpha or VAX computer, including satellites, in the cluster. |

### 2.3.5 Calculating Cluster Votes

The quorum algorithm operates as follows:

| Step | Action |
|---|---|
| **1** | When nodes in the OpenVMS Cluster boot, the connection manager uses the largest value for EXPECTED_VOTES of all systems present to derive an **estimated quorum** value according to the following formula:<br>`Estimated quorum = (EXPECTED_VOTES + 2)/2  \| Rounded down` |
| **2** | During a state transition (whenever a node enters or leaves the cluster or when a quorum disk is recognized), the connection manager dynamically computes the cluster quorum value to be the *maximum* of the following:<br>• The current cluster quorum value (calculated during the last cluster transition).<br>• Estimated quorum, as described in step 1.<br>• The value calculated from the following formula, where the VOTES system parameter is the total votes held by all cluster members: `QUORUM = (VOTES + 2)/2 \| Rounded down`<br>**Note:** Quorum disks are discussed in Section 2.3.7. |
| **3** | The connection manager compares the cluster votes value to the cluster quorum value and determines what action to take based on the following conditions: |

| WHEN... | THEN... |
|---|---|
| The total number of cluster votes is equal to at least the quorum value | The OpenVMS Cluster system continues running. |
| The current number of cluster votes drops below the quorum value (because of computers leaving the cluster) | The remaining OpenVMS Cluster members suspend all process activity and all I/O operations to cluster-accessible disks and tapes until sufficient votes are added (that is, enough computers have joined the OpenVMS Cluster) to bring the total number of votes to a value greater than or equal to quorum. |

**Note:** When a node leaves the OpenVMS Cluster system, the connection manager does not decrease the cluster quorum value. In fact, the connection manager never decreases the cluster quorum value, it only increases it, unless the REMOVE NODE option was selected during shutdown. However, system managers can decrease the value according to the instructions in Section 10.12.2.

### 2.3.6 Example

Consider a cluster consisting of three computers, each computer having its VOTES parameter set to 1 and its EXPECTED_VOTES parameter set to 3. The connection manager dynamically computes the cluster quorum value to be 2 (that is, (3 + 2)/2). In this example, any two of the three computers constitute a quorum and can run in the absence of the third computer. No single computer can constitute a quorum by itself. Therefore, there is no way the three OpenVMS Cluster computers can be partitioned and run as two independent clusters.

### 2.3.7 Quorum Disk

A cluster system manager can designate a disk a **quorum disk**. The quorum disk acts as a *virtual* cluster member whose purpose is to add one vote to the total cluster votes. By establishing a quorum disk, you can increase the availability of a two-node cluster; such configurations can maintain quorum in the event of failure of either the quorum disk or one node, and continue operating.

**Note:** Setting up a quorum disk is recommended only for OpenVMS Cluster configurations with two nodes. A quorum disk is neither necessary nor recommended for configurations with more than two nodes.

For example, assume an OpenVMS Cluster configuration with many satellites (that have no votes) and two nonsatellite systems (each having one vote) that downline load the satellites. Quorum is calculated as follows:

```
(EXPECTED VOTES + 2)/2 = (2 + 2)/2 = 2
```

Because there is no quorum disk, if either nonsatellite system departs from the cluster, only one vote remains and cluster quorum is lost. Activity will be blocked throughout the cluster until quorum is restored.

However, if the configuration includes a quorum disk (adding one vote to the total cluster votes), and the EXPECTED_VOTES parameter is set to 3 on each node, then quorum will still be 2 even if one of the nodes leaves the cluster. Quorum is calculated as follows:

```
(EXPECTED VOTES + 2)/2 = (3 + 2)/2 = 2
```

**Rules:** Each OpenVMS Cluster system can include only one quorum disk. At least one computer must have a direct (not served) connection to the quorum disk:

- Any computers that have a direct, active connection to the quorum disk or that have the potential for a direct connection should be enabled as **quorum disk watchers**.
- Computers that cannot access the disk directly must rely on the quorum disk watchers for information about the status of votes contributed by the quorum disk.

**Reference:** For more information about enabling a quorum disk, see Section 8.2.4. Section 8.3.2 describes removing a quorum disk.

### 2.3.8 Quorum Disk Watcher

To enable a computer as a quorum disk watcher, use one of the following methods:

| Method | Perform These Steps |
|---|---|
| Run the CLUSTER_CONFIG.COM procedure (described in Chapter 8) | Invoke the procedure and:<br>1. Select the CHANGE option.<br>2. From the CHANGE menu, select the item labeled "Enable a quorum disk on the local computer".<br>3. At the prompt, supply the quorum disk device name.<br>The procedure uses the information you provide to update the values of the DISK_QUORUM and QDSKVOTES system parameters. |
| Respond YES when the OpenVMS installation procedure asks whether the cluster will contain a quorum disk (described in Chapter 4) | During the installation procedure:<br>1. Answer Y when the the procedure asks whether the cluster will contain a quorum disk.<br>2. At the prompt, supply the quorum disk device name.<br>The procedure uses the information you provide to update the values of the DISK_QUORUM and QDSKVOTES system parameters. |
| Edit the MODPARAMS or AGEN$ files (described in Chapter 8) | Edit the following parameters:<br>• DISK_QUORUM: Specify the quorum disk name, in ASCII, as a value for the DISK_QUORUM system parameter.<br>• QDSKVOTES: Set an appropriate value for the QDSKVOTES parameter. This parameter specifies the number of votes contributed to the cluster votes total by a quorum disk. The number of votes contributed by the quorum disk is equal to the smallest value of the QDSKVOTES parameter on any quorum disk watcher. |

**Hint:** If only one quorum disk watcher has direct access to the quorum disk, then remove the disk and give its votes to the node.

### 2.3.9 Rules for Specifying Quorum

For the quorum disk's votes to be counted in the total cluster votes, the following conditions must be met:

- On all computers capable of becoming watchers, you must specify the same *physical* device name as a value for the DISK_QUORUM system parameter. The remaining computers (which must have a blank value for DISK_QUORUM) recognize the name specified by the first quorum disk watcher with which they communicate.
- At least one quorum disk watcher must have a direct, active connection to the quorum disk.
- The disk must contain a valid format file named QUORUM.DAT in the master file directory. The QUORUM.DAT file is created automatically after a system specifying a quorum disk has booted into the cluster for the first time. This file is used on subsequent reboots.

  **Note:** The file is not created if the system parameter STARTUP_P1 is set to MIN.
- To permit recovery from failure conditions, the quorum disk must be mounted by all disk watchers.

- The OpenVMS Cluster can include only one quorum disk.
- The quorum disk cannot be a member of a shadow set.

**Hint:** By increasing the quorum disk's votes to one less than the total votes from both systems (and by increasing the value of the EXPECTED_VOTES system parameter by the same amount), you can boot and run the cluster with only one node.

## 2.4 State Transitions

OpenVMS Cluster state transitions occur when a computer joins or leaves an OpenVMS Cluster system and when the cluster recognizes a quorum disk state change. The connection manager controls these events to ensure the preservation of data integrity throughout the cluster.

A state transition's duration and effect on users (applications) are determined by the reason for the transition, the configuration, and the applications in use.

### 2.4.1 Adding a Member

Every transition goes through one or more phases, depending on whether its cause is the addition of a new OpenVMS Cluster member or the failure of a current member.

Table 2–2 describes the phases of a transition caused by the addition of a new member.

**Table 2–2 Transitions Caused by Adding a Cluster Member**

| Phase | Description |
|---|---|
| New member detection | Early in its boot sequence, a computer seeking membership in an OpenVMS Cluster system sends messages to current members asking to join the cluster. The first cluster member that receives the membership request acts as the new computer's advocate and proposes reconfiguring the cluster to include the computer in the cluster. While the new computer is booting, no applications are affected.<br><br>**Note:** The connection manager will not allow a computer to join the OpenVMS Cluster system if the node's value for EXPECTED_VOTES would readjust quorum higher than calculated votes to cause the OpenVMS Cluster to suspend activity. |
| Reconfiguration | During a configuration change due to a computer being added to an OpenVMS Cluster, all current OpenVMS Cluster members must establish communications with the new computer. Once communications are established, the new computer is admitted to the cluster. In some cases, the lock database is rebuilt. |

### 2.4.2 Losing a Member

Table 2–3 describes the phases of a transition caused by the failure of a current OpenVMS Cluster member.

**Table 2–3 Transitions Caused by Loss of a Cluster Member**

| Cause | Description |
|---|---|
| Failure detection | The duration of this phase depends on the cause of the failure and on how the failure is detected.<br>During normal cluster operation, messages sent from one computer to another are acknowledged when received. |

| IF... | THEN... |
|---|---|
| A message is not acknowledged within a period determined by OpenVMS Cluster communications software | The repair attempt phase begins. |
| A cluster member is shut down or fails | The operating system causes datagrams to be sent from the computer shutting down to the other members. These datagrams state the computer's intention to sever communications and to stop sharing resources. The failure detection and repair attempt phases are bypassed, and the reconfiguration phase begins immediately. |

| | |
|---|---|
| Repair attempt | If the virtual circuit to an OpenVMS Cluster member is broken, attempts are made to repair the path. Repair attempts continue for an interval specified by the PAPOLLINTERVAL system parameter. (System managers can adjust the value of this parameter to suit local conditions.) Thereafter, the path is considered irrevocably broken, and steps must be taken to reconfigure the OpenVMS Cluster system so that all computers can once again communicate with each other and so that computers that cannot communicate are removed from the OpenVMS Cluster. |
| Reconfiguration | If a cluster member is shut down or fails, the cluster must be reconfigured. One of the remaining computers acts as coordinator and exchanges messages with all other cluster members to determine an optimal cluster configuration with the most members and the most votes. This phase, during which all user (application) activity is blocked, usually lasts less than 3 seconds, although the actual time depends on the configuration. |

(continued on next page)

**Table 2–3 (Cont.) Transitions Caused by Loss of a Cluster Member**

| Cause | Description |
|---|---|
| OpenVMS Cluster system recovery | Recovery includes the following stages, some of which can take place in parallel: |

| Stage | Action |
|---|---|
| I/O completion | When a computer is removed from the cluster, OpenVMS Cluster software ensures that all I/O operations that are started prior to the transition complete before I/O operations that are generated after the transition. This stage usually has little or no effect on applications. |
| Lock database rebuild | Because the lock database is distributed among all members, some portion of the database might need rebuilding. A rebuild is performed as follows: |

| WHEN... | THEN... |
|---|---|
| A computer leaves the OpenVMS Cluster | A rebuild is always performed. |
| A computer is added to the OpenVMS Cluster | A rebuild is performed when the LOCKDIRWT system parameter is greater than 1. |

**Caution:** Setting the LOCKDIRWT system parameter to different values on the same model or type of computer can cause the distributed lock manager to use the computer with the higher value. This could cause undue resource usage on that computer.

| | |
|---|---|
| Disk mount verification | This stage occurs only when the failure of a voting member causes quorum to be lost. To protect data integrity, all I/O activity is blocked until quorum is regained. Mount verification is the mechanism used to block I/O during this phase. |
| Quorum disk votes validation | If, when a computer is removed, the remaining members can determine that it has shut down or failed, the votes contributed by the quorum disk are included without delay in quorum calculations that are performed by the remaining members. However, if the quorum watcher cannot determine that the computer has shut down or failed (for example, if a console halt, power failure, or communications failure has occurred), the votes are not included for a period (in seconds) equal to four times the value of the QDSKINTERVAL system parameter. This period is sufficient to determine that the failed computer is no longer using the quorum disk. |
| Disk rebuild | If the transition is the result of a computer rebooting after a failure, the disks are marked as improperly dismounted.<br>**Reference:** See Sections 6.5.5 and 6.5.6 for information about rebuilding disks. |

| | |
|---|---|
| Application recovery | When you assess the effect of a state transition on application users, consider that the application recovery phase includes activities such as replaying a journal file, cleaning up recovery units, and users logging in again. |

## 2.5 OpenVMS Cluster Membership

OpenVMS Cluster systems based on LAN use a cluster group number and a cluster password to allow multiple independent OpenVMS Cluster systems to coexist on the same extended LAN and to prevent accidental access to a cluster by unauthorized computers.

### 2.5.1 Cluster Group Number

The **cluster group number** uniquely identifies each OpenVMS Cluster system on a LAN. This number must be from 1 to 4095 or from 61440 to 65535.

**Rule:** If you plan to have more than one OpenVMS Cluster system on a LAN, you must coordinate the assignment of cluster group numbers among system managers.

**Note:** OpenVMS Cluster systems operating on CI and DSSI do not use cluster group numbers and passwords.

### 2.5.2 Cluster Password

The **cluster password** prevents an unauthorized computer using the cluster group number, from joining the cluster. The password must be from 1 to 31 alphanumeric characters in length, including dollar signs ( $ ) and underscores ( _ ).

### 2.5.3 Location

The cluster group number and cluster password are maintained in the cluster authorization file, SYS$COMMON:[SYSEXE]CLUSTER_AUTHORIZE.DAT. This file is created during installation of the operating system if you indicate that you want to set up a cluster that utilizes the LAN. The installation procedure then prompts you for the cluster group number and password.

**Note:** If you convert an OpenVMS Cluster that uses only the CI or DSSI interconnect to one that includes a LAN interconnect, the SYS$COMMON:[SYSEXE]CLUSTER_AUTHORIZE.DAT file is created when you execute the CLUSTER_CONFIG.COM command procedure, as described in Chapter 8.

**Reference:** For information about OpenVMS Cluster group data in the CLUSTER_AUTHORIZE.DAT file, see Sections 8.4 and 10.9.

### 2.5.4 Example

If all nodes in the OpenVMS Cluster do not have the same cluster password, an error report similar to the following is logged in the error log file.

```
V A X / V M S        SYSTEM ERROR REPORT         COMPILED 30-JAN-1994 15:38:03
                                                                  PAGE  19.

****************************** ENTRY     161. ******************************
ERROR SEQUENCE 24.                          LOGGED ON:        SID 12000003
DATE/TIME 30-JAN-1994 15:35:47.94                          SYS_TYPE 04010002
SYSTEM UPTIME: 5 DAYS 03:46:21
SCS NODE: DAISIE                                            VAX/VMS V6.0

DEVICE ATTENTION  KA46  CPU FW REV# 3.  CONSOLE FW REV# 0.1

NI-SCS SUB-SYSTEM, DAISIE$PEA0:

      INVALID CLUSTER PASSWORD RECEIVED
```

```
STATUS          00000000
                00000000
DATALINK UNIT       0001
DATALINK NAME   41534503
                00000000
                00000000
                00000000
                                DATALINK NAME = ESA1:
REMOTE NODE     554C4306
                00203132
                00000000
                00000000
                                REMOTE NODE = CLU21
REMOTE ADDR     000400AA
                    FC15
                                ETHERNET ADDR = AA-00-04-00-15-FC
LOCAL ADDR      000400AA
                    4D34
                                ETHERNET ADDR = AA-00-04-00-34-4D
ERROR CNT           0001
                                1. ERROR OCCURRENCES THIS ENTRY
UCB$W_ERRCNT        0003
                                3. ERRORS THIS UNIT
```

## 2.6 Synchronizing Cluster Functions by the Distributed Lock Manager

The **distributed lock manager** is an OpenVMS feature for synchronizing functions required by the distributed file system, the distributed job controller, device allocation, user-written OpenVMS Cluster applications, and other OpenVMS products and software components.

The distributed lock manager uses the connection manager and SCS to communicate information between OpenVMS Cluster computers.

### 2.6.1 Distributed Lock Manager Functions

The functions of the distributed lock manager include the following:

- Synchronizes access to shared clusterwide resources, including:
  - Devices
  - Files
  - Records in files
  - Any user-defined resources, such as databases and memory

  Each resource is managed clusterwide by an OpenVMS Cluster computer.
- Implements the $ENQ and $DEQ system services to provide clusterwide synchronization of access to resources by allowing the locking and unlocking of resource names.

  **Reference:** For detailed information about system services, refer to the *OpenVMS System Services Reference Manual*.
- Queues process requests for access to a locked resource. This queuing mechanism allows processes to be put into a wait state until a particular resource is available. As a result, cooperating processes can synchronize their access to shared objects, such as files and records.
- Releases all locks that an OpenVMS Cluster computer holds if the computer fails. This mechanism allows processing to continue on the remaining computers.

- Supports clusterwide deadlock detection.

### 2.6.2 System Management of the Lock Manager

The lock manager is fully automated and usually requires no explicit system management. However, the LOCKDIRWT system parameter can be used to adjust how control of lock resource trees is distributed across the cluster.

The node that controls a lock resource tree is called the resource master. Each resource tree may be mastered by a different node.

For most configurations, large computers and boot nodes perform optimally when LOCKDIRWT is set to 1 and satellite nodes have LOCKDIRWT set to 0. These values are set automatically by the CLUSTER_CONFIG.COM procedure.

In some circumstances, you may want to change the values of the LOCKDIRWT across the cluster to control which nodes master resource trees. The following list describes how the value of the LOCKDIRWT system parameter affects resource tree mastership:

- If multiple nodes have locks on a resource tree, the tree is mastered by the node with the highest value for LOCKDIRWT, regardless of actual locking rates.
- If multiple nodes with the same LOCKDIRWT value have locks on a resource, the tree is mastered by the node with the highest locking rate on that tree.
- Note that if only one node has locks on a resource tree, it becomes the master of the tree, regardless of the LOCKDIRWT value.

Thus, using varying values for the LOCKDIRWT system parameter, you can implement a resource tree mastering policy that is priority based. Using equal values for the LOCKDIRWT system parameter, you can implement a resource tree mastering policy that is activity based. If necessary, a combination of priority-based and activity-based remastering can be used.

### 2.6.3 Large-Scale Locking Applications

The Enqueue process limit (ENQLM), which is set in the SYSUAF.DAT file and which controls the number of locks that a process can own, can be adjusted to meet the demands of large scale databases and other server applications.

Prior to OpenVMS Version 7.1, the limit was 32767. This limit was removed to enable the efficient operation of large scale databases and other server applications. A process can now own up to 16,776,959 locks, the architectural maximum. By setting ENQLM in SYSUAF.DAT to 32767 (using the Authorize utility), the lock limit is automatically extended to the maximum of 16,776,959 locks. $CREPRC can pass large quotas to the target process if it is initialized from a process with the SYSUAF Enqlm quota of 32767.

**Reference:** See the *OpenVMS Programming Concepts Manual* for additional information about the distributed lock manager and resource trees. See the *OpenVMS System Manager's Manual* for more information about Enqueue Quota.

## 2.7 Resource Sharing

Resource sharing in an OpenVMS Cluster system is enabled by the distributed file system, RMS, and the distributed lock manager.

### 2.7.1 Distributed File System

The OpenVMS Cluster **distributed file system** allows all computers to share mass storage and files. The distributed file system provides the same access to disks, tapes, and files across the OpenVMS Cluster that is provided on a standalone computer.

### 2.7.2 RMS and Distributed Lock Manager

The distributed file system and OpenVMS Record Management Services (RMS) use the distributed lock manager to coordinate clusterwide file access. RMS files can be shared to the record level.

Any disk or tape can be made available to the entire OpenVMS Cluster system. The storage devices can be:

- Connected to an HSC, HSJ, HSD, HSG, HSZ, DSSI, or SCSI subsystem
- A local device that is served to the OpenVMS Cluster

All cluster-accessible devices appear as if they are connected to every computer.

## 2.8 Disk Availability

Locally connected disks can be served across an OpenVMS Cluster by the MSCP server.

### 2.8.1 MSCP Server

The **MSCP server** makes locally connected disks, including the following, available across the cluster:

- DSA disks local to OpenVMS Cluster members using SDI
- HSC and HSJ disks in an OpenVMS Cluster using mixed interconnects
- ISE and HSD disks in an OpenVMS Cluster using mixed interconnects
- SCSI and HSZ disks
- FC and HSG disks
- Disks on boot servers and disk servers located anywhere in the OpenVMS Cluster

In conjunction with the disk class driver (DUDRIVER), the MSCP server implements the storage server portion of the MSCP protocol on a computer, allowing the computer to function as a storage controller. The MSCP protocol defines conventions for the format and timing of messages sent and received for certain families of mass storage controllers and devices designed by Compaq. The MSCP server decodes and services MSCP I/O requests sent by remote cluster nodes.

**Note:** The MSCP server is not used by a computer to access files on locally connected disks.

### 2.8.2 Device Serving

Once a device is set up to be served:

- Any cluster member can submit I/O requests to it.
- The local computer can decode and service MSCP I/O requests sent by remote OpenVMS Cluster computers.

### 2.8.3 Enabling the MSCP Server

The MSCP server is controlled by the MSCP_LOAD and MSCP_SERVE_ALL system parameters. The values of these parameters are set initially by answers to questions asked during the OpenVMS installation procedure (described in Section 8.4), or during the CLUSTER_CONFIG.COM procedure (described in Chapter 8).

The default values for these parameters are as follows:

- MSCP is not loaded on satellites.
- MSCP is loaded on boot server and disk server nodes.

**Reference:** See Section 6.3 for more information about setting system parameters for MSCP serving.

## 2.9 Tape Availability

Locally connected tapes can be served across an OpenVMS Cluster by the TMSCP server.

### 2.9.1 TMSCP Server

The **TMSCP server** makes locally connected tapes, including the following, available across the cluster:

- HSC and HSJ tapes
- ISE and HSD tapes
- SCSI tapes

The TMSCP server implements the TMSCP protocol, which is used to communicate with a controller for TMSCP tapes. In conjunction with the tape class driver (TUDRIVER), the TMSCP protocol is implemented on a processor, allowing the processor to function as a storage controller.

The processor submits I/O requests to locally accessed tapes, and accepts the I/O requests from any node in the cluster. In this way, the TMSCP server makes locally connected tapes available to all nodes in the cluster. The TMSCP server can also make HSC tapes and DSSI ISE tapes accessible to OpenVMS Cluster satellites.

### 2.9.2 Enabling the TMSCP Server

The TMSCP server is controlled by the TMSCP_LOAD system parameter. The value of this parameter is set initially by answers to questions asked during the OpenVMS installation procedure (described in Section 4.2.3) or during the CLUSTER_CONFIG.COM procedure (described in Section 8.4). By default, the setting of the TMSCP_LOAD parameter does not load the TMSCP server and does not serve any tapes.

## 2.10 Queue Availability

The **distributed job controller** makes queues available across the cluster in order to achieve the following:

| Function | Description |
|---|---|
| Permit users on any OpenVMS Cluster computer to submit batch and print jobs to queues that execute on any computer in the OpenVMS Cluster | Users can submit jobs to any queue in the cluster, provided that the necessary mass storage volumes and peripheral devices are accessible to the computer on which the job executes. |
| Distribute the batch and print processing work load over OpenVMS Cluster nodes | System managers can set up generic batch and print queues that distribute processing work loads among computers. The distributed job controller directs batch and print jobs either to the execution queue with the lowest ratio of jobs-to-queue limit or to the next available printer. |

The job controller uses the distributed lock manager to signal other computers in the OpenVMS Cluster to examine the batch and print queue jobs to be processed.

### 2.10.1 Controlling Queues

To control queues, you use one or several queue managers to maintain a clusterwide queue database that stores information about queues and jobs.

**Reference:** For detailed information about setting up OpenVMS Cluster queues, see Chapter 7.

# 3

# OpenVMS Cluster Interconnect Configurations

This chapter provides an overview of various types of OpenVMS Cluster configurations and the ways they are interconnected.

**References:** For definitive information about supported OpenVMS Cluster configurations, refer to:

- OpenVMS Cluster Software *Software Product Description* (SPD 29.78.*xx*)
- *Guidelines for OpenVMS Cluster Configurations*

## 3.1 Overview

All Alpha and VAX nodes in any type of OpenVMS Cluster must have direct connections to all other nodes. Sites can choose to use one or more of the following interconnects:

- LANs
  - ATM
  - Ethernet (10/100 and Gigabit Ethernet)
  - FDDI
- CI
- DSSI
- MEMORY CHANNEL
- SCSI (requires a second interconnect for node-to-node [SCS] communications)
- Fibre Channel (requires a second interconnect for node-to-node [SCS] communications)

Processing needs and available hardware resources determine how individual OpenVMS Cluster systems are configured. The configuration discussions in this chapter are based on these physical interconnects.

## 3.2 OpenVMS Cluster Systems Interconnected by CI

The CI was the first interconnect used for OpenVMS Cluster communications. The CI supports the exchange of information among VAX and Alpha nodes, and HSC and HSJ nodes at the rate of 70 megabits per second on two paths.

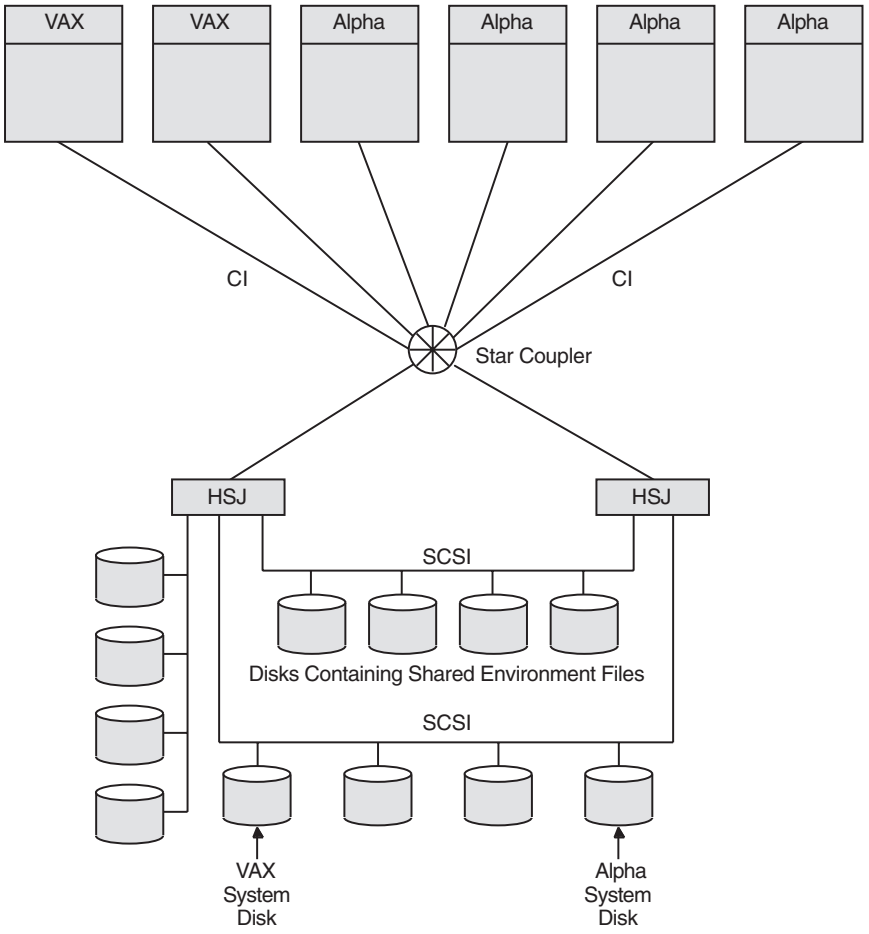
### 3.2.1 Design

The CI is designed for access to storage and for reliable host-to-host communication. CI is a high-performance, highly available way to connect Alpha and VAX nodes to disk and tape storage devices and to each other. An OpenVMS Cluster system based on the CI for cluster communications uses star couplers as common connection points for computers, and HSC and HSJ subsystems.

### 3.2.2 Example

Figure 3–1 shows how the CI components are typically configured.

**Figure 3–1 OpenVMS Cluster Configuration Based on CI**




**Note:** If you want to add workstations to a CI OpenVMS Cluster system, you must utilize an additional type of interconnect, such as Ethernet or FDDI, in the configuration. Workstations are typically configured as satellites in an OpenVMS Cluster system (see Section 3.4.4).

**Reference:** For instructions on adding satellites to an existing CI OpenVMS Cluster system, refer to Section 8.2.

### 3.2.3 Star Couplers

What appears to be a single point of failure in the CI configuration in Figure 3–1 is the star coupler that connects all the CI lines. In reality, the star coupler is not a single point of failure because there are actually two star couplers in every cabinet.

Star couplers are also immune to power failures because they contain no powered components but are constructed as sets of high-frequency pulse transformers. Because they do no processing or buffering, star couplers also are not I/O throughput bottlenecks. They operate at the full-rated speed of the CI cables. However, in very heavy I/O situations, exceeding CI bandwidth may require multiple star couplers.

## 3.3 OpenVMS Cluster Systems Interconnected by DSSI

The DIGITAL Storage Systems Interconnect (DSSI) is a medium-bandwidth interconnect that Alpha and VAX nodes can use to access disk and tape peripherals. Each peripheral is an integrated storage element (ISE) that contains its own controller and its own MSCP server that works in parallel with the other ISEs on the DSSI.

### 3.3.1 Design

Although the DSSI is designed primarily to access disk and tape storage, it has proven an excellent way to connect small numbers of nodes using the OpenVMS Cluster protocols. Each DSSI port connects to a single DSSI bus. As in the case of the CI, several DSSI ports can be connected to a node to provide redundant paths between nodes. However, unlike CI, DSSI does not provide redundant paths.

### 3.3.2 Availability

OpenVMS Cluster configurations using ISE devices and the DSSI bus offer high availability, flexibility, growth potential, and ease of system management.

DSSI nodes in an OpenVMS Cluster configuration can access a common system disk and all data disks directly on a DSSI bus and serve them to satellites. Satellites (and users connected through terminal servers) can access any disk through any node designated as a boot server. If one of the boot servers fails, applications on satellites continue to run because disk access fails over to the other server. Although applications running on nonintelligent devices, such as terminal servers, are interrupted, users of terminals can log in again and restart their jobs.

### 3.3.3 Guidelines

Generic configuration guidelines for DSSI OpenVMS Cluster systems are as follows:

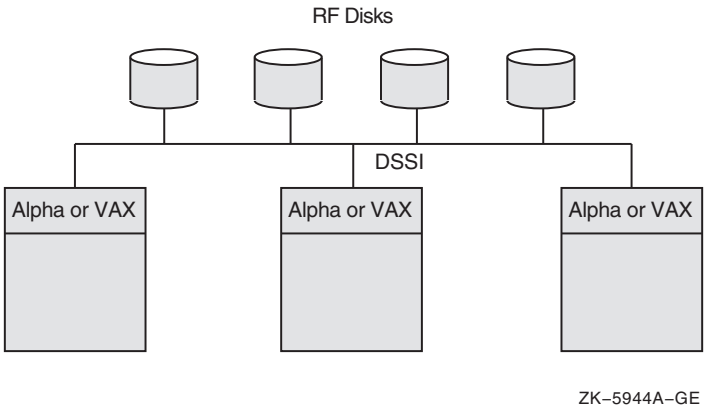
- Currently, a total of four Alpha and/or VAX nodes can be connected to a common DSSI bus.
- Multiple DSSI buses can operate in an OpenVMS Cluster configuration, thus dramatically increasing the amount of storage that can be configured into the system.

**References:** Some restrictions apply to the type of CPUs and DSSI I/O adapters that can reside on the same DSSI bus. Consult your service representative or see the OpenVMS Cluster Software *Software Product Description* (SPD) for complete and up-to-date configuration details about DSSI OpenVMS Cluster systems.

### 3.3.4 Example

Figure 3–2 shows a typical DSSI configuration.

**Figure 3–2 DSSI OpenVMS Cluster Configuration**




## 3.4 OpenVMS Cluster Systems Interconnected by LANs

The Ethernet (10/100 and Gigabit), FDDI, and ATM interconnects are industry-standard local area networks (LANs) that are generally shared by a wide variety of network consumers. When OpenVMS Cluster systems are based on LAN, cluster communications are carried out by a port driver (PEDRIVER) that emulates CI port functions.

### 3.4.1 Design

The OpenVMS Cluster software is designed to use the Ethernet, ATM, and FDDI ports and interconnects simultaneously with the DECnet, TCP/IP, and SCS protocols. This is accomplished by allowing LAN data link software to control the hardware port. This software provides a multiplexing function so that the cluster protocols are simply another user of a shared hardware resource. See Figure 2–1 for an illustration of this concept.

### 3.4.2 Cluster Group Numbers and Cluster Passwords

A single LAN can support multiple LAN-based OpenVMS Cluster systems. Each OpenVMS Cluster is identified and secured by a unique cluster group number and a cluster password. Chapter 2 describes cluster group numbers and cluster passwords in detail.

### 3.4.3 Servers

OpenVMS Cluster computers interconnected by a LAN are generally configured as either servers or satellites. The following table describes servers.

| Server Type | Description |
|---|---|
| MOP servers | Downline load the OpenVMS boot driver to satellites by means of the Maintenance Operations Protocol (MOP). |
| Disk servers | Use MSCP server software to make their locally connected disks and any CI or DSSI connected disks available to satellites over the LAN. |
| Tape servers | Use TMSCP server software to make their locally connected tapes and any CI or DSSI connected tapes available to satellite nodes over the LAN. |
| Boot servers | A combination of a MOP server and a disk server that serves one or more Alpha or VAX system disks. Boot and disk servers make user and application data disks available across the cluster. These servers should be the most powerful computers in the OpenVMS Cluster and should use the highest-bandwidth LAN adapters in the cluster. Boot servers must always run the MSCP server software. |

### 3.4.4 Satellites

**Satellites** are computers without a local system disk. Generally, satellites are consumers of cluster resources, although they can also provide facilities for disk serving, tape serving, and batch processing. If satellites are equipped with local disks, they can enhance performance by using such local disks for paging and swapping.

Satellites are booted remotely from a boot server (or from a MOP server and a disk server) serving the system disk. Section 3.4.5 describes MOP and disk server functions during satellite booting.

**Note:** An Alpha system disk can be mounted as a data disk on a VAX computer and, with proper MOP setup, can be used to boot Alpha satellites. Similarly, a VAX system disk can be mounted on an Alpha computer and, with the proper MOP setup, can be used to boot VAX satellites.

**Reference:** Cross-architecture booting is described in Section 10.5.

### 3.4.5 Satellite Booting

When a satellite requests an operating system load, a MOP server for the appropriate OpenVMS Alpha or OpenVMS VAX operating system sends a bootstrap image to the satellite that allows the satellite to load the rest of the operating system from a disk server and join the cluster. The sequence of actions during booting is described in Table 3–1.

**Table 3–1 Satellite Booting Process**

| Step | Action | Comments |
|---|---|---|
| **1** | Satellite requests MOP service. | This is the original boot request that a satellite sends out across the network. Any node in the OpenVMS Cluster that has MOP service enabled and has the LAN address of the particular satellite node in its database can become the MOP server for the satellite. |
| **2** | MOP server loads the Alpha or VAX system. | ‡The MOP server responds to an Alpha satellite boot request by downline loading the SYS$SYSTEM:APB.EXE program along with the required parameters.<br>†The MOP server responds to a VAX satellite boot request by downline loading the SYS$SHARE:NISCS_LOAD.EXE program along with the required parameters.<br>For Alpha and VAX computers, Some of these parameters include:<br>• System disk name<br>• Root number of the satellite |
| **3** | Satellite finds additional parameters located on the system disk and root. | The satellite finds OpenVMS Cluster system parameters, such as SCSSYSTEMID, SCSNODE, and NISCS_CONV_BOOT. The satellite also finds the cluster group code and password. |
| **4** | Satellite executes the load program | The program establishes an SCS connection to a disk server for the satellite system disk and loads the SYSBOOT.EXE program. |

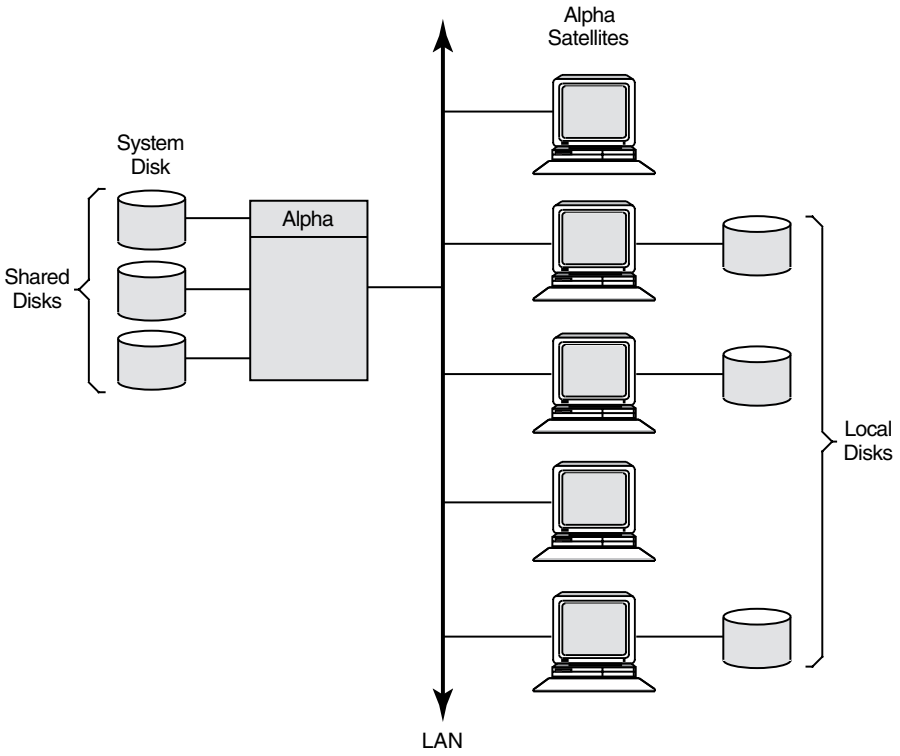
†VAX specific
‡Alpha specific

### 3.4.6 Examples

Figure 3–3 shows an OpenVMS Cluster system based on a LAN interconnect with a single Alpha server node and a single Alpha system disk.

**Note:** To include VAX satellites in this configuration, configure a VAX system disk on the Alpha server node following the instructions in Section 10.5.

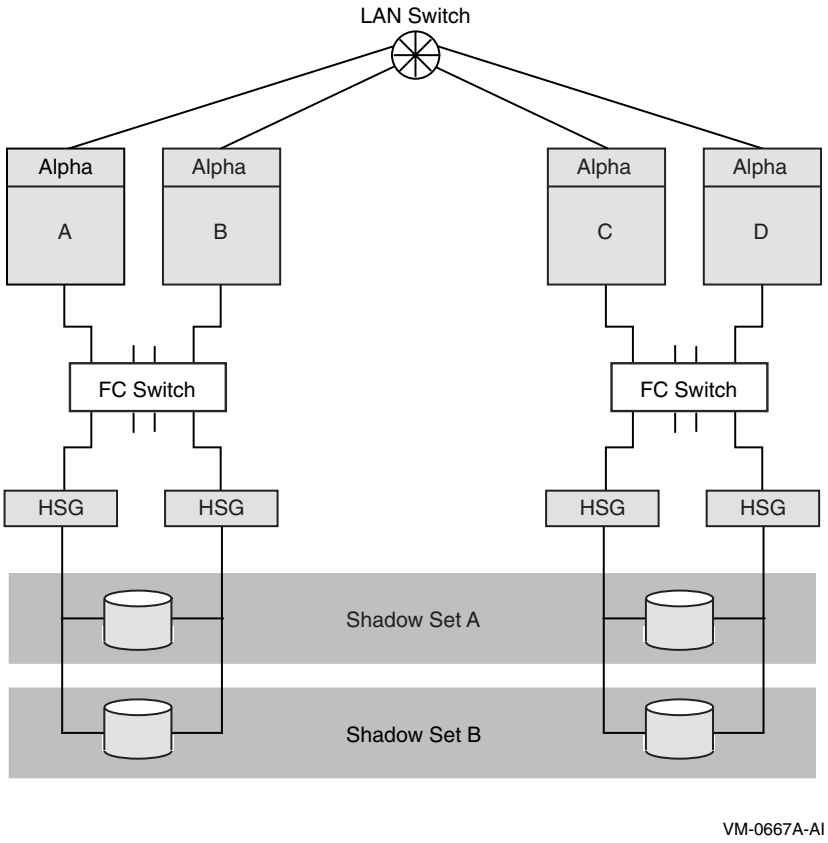
**Figure 3–3 LAN OpenVMS Cluster System with Single Server Node and System Disk**




In Figure 3–3, the server node (and its system disk) is a single point of failure. If the server node fails, the satellite nodes cannot access any of the shared disks including the system disk. Note that some of the satellite nodes have locally connected disks. If you convert one or more of these into system disks, satellite nodes can boot from their own local system disk.

Figure 3–4 shows an example of an OpenVMS Cluster system that uses LAN and Fibre Channel interconnects.

**Figure 3–4 LAN and Fibre Channel OpenVMS Cluster System: Sample Configuration**

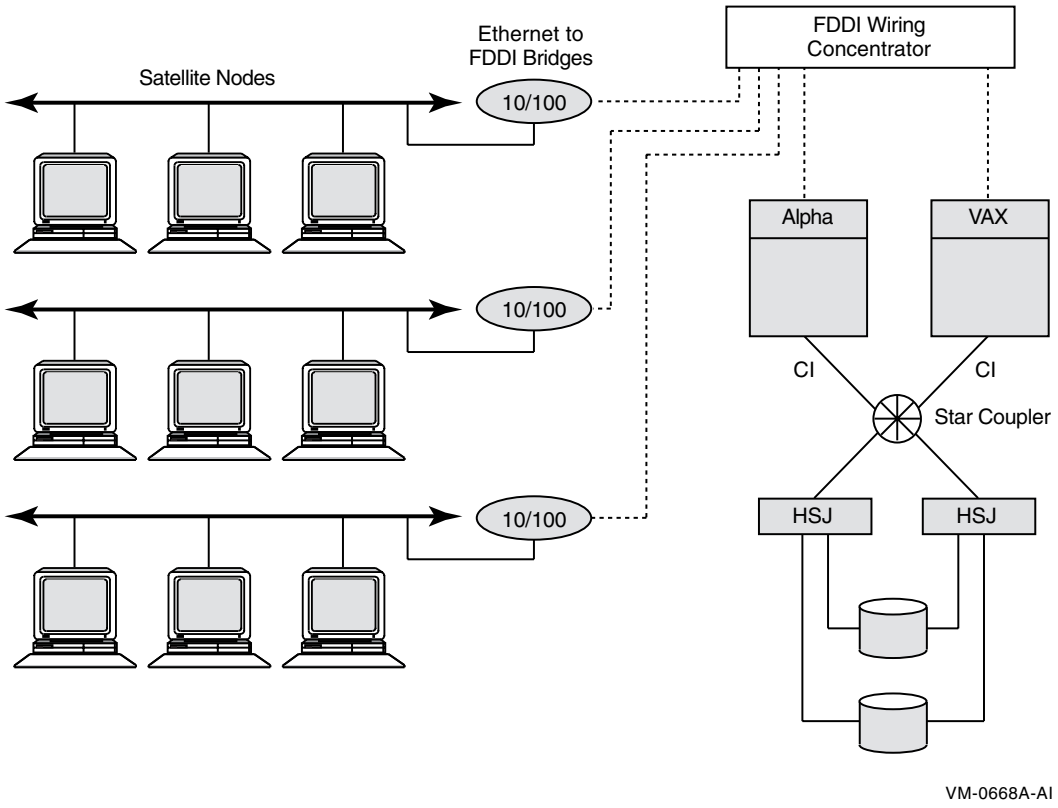



The LAN connects nodes A and B with nodes C and D into a single OpenVMS Cluster system.

In Figure 3–4, Volume Shadowing for OpenVMS is used to maintain key data storage devices in identical states (shadow sets A and B). Any data on the shadowed disks written at one site will also be written at the other site. However, the benefits of high data availability must be weighed against the performance overhead required to use the MSCP server to serve the shadow set over the cluster interconnect.

Figure 3–5 illustrates how FDDI can be configured with Ethernet from the bridges to the server CPU nodes. This configuration can increase overall throughput. OpenVMS Cluster systems that have heavily utilized Ethernet segments can replace the Ethernet backbone with a faster LAN to alleviate the performance bottleneck that can be caused by the Ethernet.

**Figure 3–5 FDDI in Conjunction with Ethernet in an OpenVMS Cluster System**




**Comments:**

- Each satellite LAN segment could be in a different building or town because of the longer distances allowed by FDDI.
- The longer distances provided by FDDI permit you to create new OpenVMS Cluster systems in your computing environment. The large nodes on the right could have replaced server nodes that previously existed on the individual Ethernet segments.
- The VAX and Alpha computers have CI connections for storage. Currently, no storage controllers connect directly to FDDI. CPU nodes connected to FDDI must have local storage or access to storage over another interconnect.

  If an OpenVMS Cluster system has more than one FDDI-connected node, then those CPU nodes will probably use CI or DSSI connections for storage. The VAX and Alpha computers, connected by CI in Figure 3–5, are considered a lobe of the OpenVMS Cluster system.

### 3.4.7 LAN Bridge Failover Process

The following table describes how the bridge parameter settings can affect the failover process.

| Option | Comments |
|---|---|
| Decreasing the LISTEN_TIME value allows the bridge to detect topology changes more quickly. | If you reduce the LISTEN_TIME parameter value, you should also decrease the value for the HELLO_INTERVAL bridge parameter according to the bridge-specific guidelines. However, note that decreasing the value for the HELLO_INTERVAL parameter causes an increase in network traffic. |
| Decreasing the FORWARDING_DELAY value can cause the bridge to forward packets unnecessarily to the other LAN segment. | Unnecessary forwarding can temporarily cause more traffic on both LAN segments until the bridge software determines which LAN address is on each side of the bridge. |

**Note:** If you change a parameter on one LAN bridge, you should change that parameter on all bridges to ensure that selection of a new root bridge does not change the value of the parameter. The actual parameter value the bridge uses is the value specified by the root bridge.

## 3.5 OpenVMS Cluster Systems Interconnected by MEMORY CHANNEL

MEMORY CHANNEL is a high-performance cluster interconnect technology for PCI-based Alpha systems. With the benefits of very low latency, high bandwidth, and direct memory access, MEMORY CHANNEL complements and extends the ability of OpenVMS Clusters to work as a single, virtual system. MEMORY CHANNEL is used for node-to-node cluster communications only. You use it in combination with another interconnect, such as Fibre Channel, SCSI, CI, or DSSI, that is dedicated to storage traffic.

### 3.5.1 Design

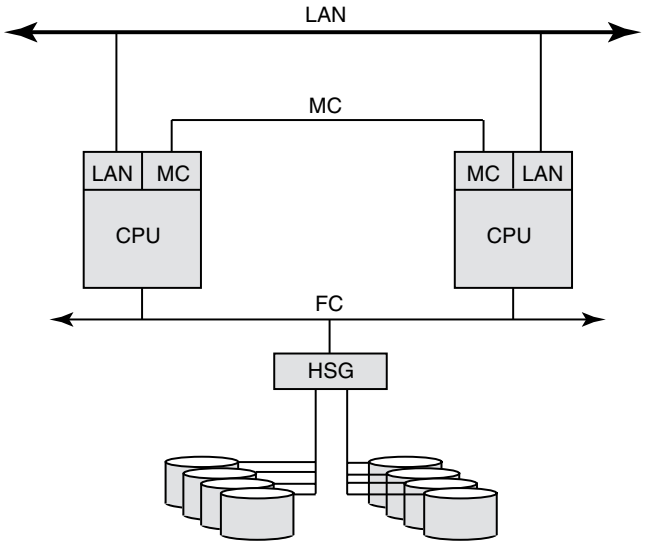
A node requires the following three hardware components to support a MEMORY CHANNEL connection:

- PCI-to MEMORY CHANNEL adapter
- Link cable (3 m or 10 feet long)
- Port in a MEMORY CHANNEL hub (except for a two-node configuration in which the cable connects just two PCI adapters)

### 3.5.2 Examples

Figure 3–6 shows a two-node MEMORY CHANNEL cluster with shared access to Fibre Channel storage and a LAN interconnect for failover.

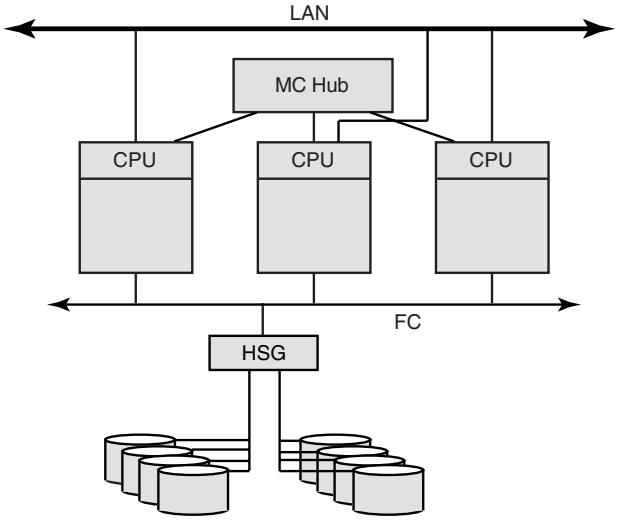
**Figure 3–6 Two-Node MEMORY CHANNEL OpenVMS Cluster Configuration**




VM-0669A-AI

A three-node MEMORY CHANNEL cluster connected by a MEMORY CHANNEL hub and also by a LAN interconnect is shown in Figure 3–7. The three nodes share access to the Fibre Channel storage. The LAN interconnect enables failover if the MEMORY CHANNEL interconnect fails.

**Figure 3–7 Three-Node MEMORY CHANNEL OpenVMS Cluster Configuration**




VM-0670A-AI

## 3.6 Multihost SCSI OpenVMS Cluster Systems

OpenVMS Cluster systems support the Small Computer Systems Interface (SCSI) as a storage interconnect. A SCSI interconnect, also called a SCSI bus, is an industry-standard interconnect that supports one or more computers, peripheral devices, and interconnecting components.

Beginning with OpenVMS Alpha Version 6.2, multiple Alpha computers can simultaneously access SCSI disks over a SCSI interconnect. Another interconnect, for example, a local area network, is required for host-to-host OpenVMS cluster communications.

### 3.6.1 Design

Beginning with OpenVMS Alpha Version 6.2-1H3, OpenVMS Alpha supports up to three nodes on a shared SCSI bus as the storage interconnect. A quorum disk can be used on the SCSI bus to improve the availability of two-node configurations. Host-based RAID (including host-based shadowing) and the MSCP server are supported for shared SCSI storage devices.

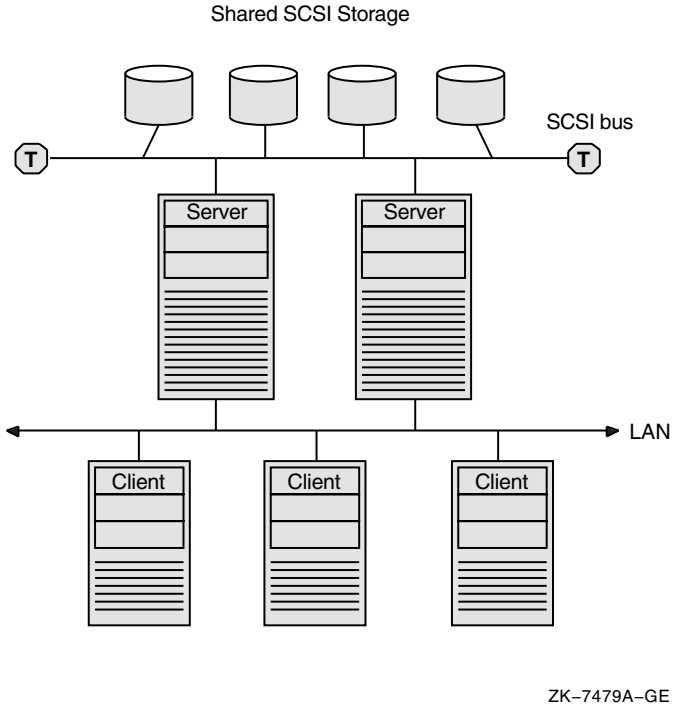
With the introduction of the SCSI hub DWZZH-05, four nodes can be supported in a SCSI multihost OpenVMS Cluster system. In order to support four nodes, the hub's fair arbitration feature must be enabled.

For a complete description of these configurations, see *Guidelines for OpenVMS Cluster Configurations*.

### 3.6.2 Examples

Figure 3–8 shows an OpenVMS Cluster configuration that uses a SCSI interconnect for shared access to SCSI devices. Note that another interconnect, a LAN in this example, is used for host-to-host communications.

**Figure 3–8 Three-Node OpenVMS Cluster Configuration Using a Shared SCSI Interconnect**




## 3.7 Multihost Fibre Channel OpenVMS Cluster Systems

OpenVMS Cluster systems support FC interconnect as a storage interconnect. Fibre Channel is an ANSI standard network and storage interconnect that offers many advantages over other interconnects, including high-speed transmission and long interconnect distances. A second interconnect is required for node-to-node communications.

### 3.7.1 Design

OpenVMS Alpha supports the Fibre Channel SAN configurations described in the latest *Compaq StorageWorks Heterogeneous Open SAN Design Reference Guide* and in the Data Replication Manager (DRM) user documentation. This configuration support includes multiswitch Fibre Channel fabrics, up to 500 meters of multimode fiber, and up to 100 kilometers of single-mode fiber. In addition, DRM configurations provide long-distance intersite links (ISLs) through the use of the Open Systems Gateway and wave division multiplexors. OpenVMS supports sharing of the fabric and the HSG storage with non-OpenVMS systems.

OpenVMS provides support for the number of hosts, switches, and storage controllers specified in the StorageWorks documentation. In general, the number of hosts and storage controllers is limited only by the number of available fabric connections.

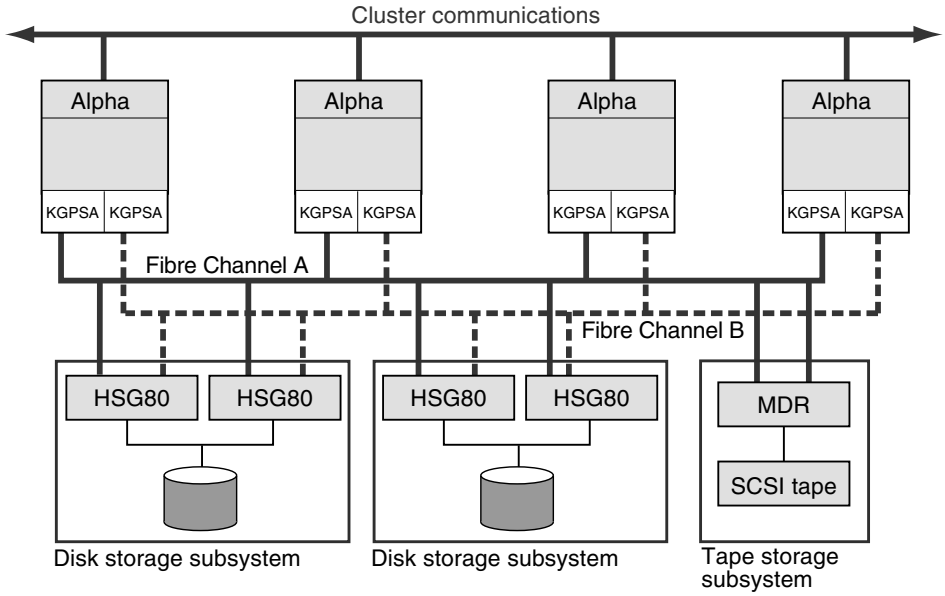
Host-based RAID (including host-based shadowing) and the MSCP server are supported for shared Fibre Channel storage devices. Multipath support is available for these configurations.

For a complete description of these configurations, see *Guidelines for OpenVMS Cluster Configurations*.

### 3.7.2 Examples

Figure 3–9 shows a multihost configuration with two independent Fibre Channel interconnects connecting the hosts to the storage subsystems. Note that another interconnect is used for node-to-node communications.

**Figure 3–9 Four-Node OpenVMS Cluster Configuration Using a Fibre Channel Interconnect**



# 4

# The OpenVMS Cluster Operating Environment

This chapter describes how to prepare the OpenVMS Cluster operating environment.

## 4.1 Preparing the Operating Environment

To prepare the cluster operating environment, there are a number of steps you perform on the first OpenVMS Cluster node before configuring other computers into the cluster. The following table describes these tasks.

| Task | Section |
|---|---|
| Check all hardware connections to computer, interconnects, and devices. | Described in the appropriate hardware documentation. |
| Verify that all microcode and hardware is set to the correct revision levels. | Contact your support representative. |
| Install the OpenVMS operating system. | Section 4.2 |
| Install all software licenses, including OpenVMS Cluster licenses. | Section 4.3 |
| Install layered products. | Section 4.4 |
| Configure and start LANCP or DECnet for satellite booting | Section 4.5 |

## 4.2 Installing the OpenVMS Operating System

Only one OpenVMS operating system version can exist on a system disk. Therefore, when installing or upgrading the OpenVMS operating systems:

- Install the OpenVMS Alpha operating system on each Alpha system disk
- Install the OpenVMS VAX operating system on each VAX system disk

### 4.2.1 System Disks

A system disk is one of the few resources that cannot be shared between Alpha and VAX systems. However, an Alpha system disk can be mounted as a data disk on a VAX computer and, with MOP configured appropriately, can be used to boot Alpha satellites. Similarly, a VAX system disk can be mounted on an Alpha computer and, with the appropriate MOP configuration, can be used to boot VAX satellites.

**Reference:** Cross-architecture booting is described in Section 10.5.

Once booted, Alpha and VAX processors can share access to data on any disk in the OpenVMS Cluster, including system disks. For example, an Alpha system can mount a VAX system disk as a data disk and a VAX system can mount an Alpha system disk as a data disk.

**Note:** An OpenVMS Cluster running both implementations of DECnet requires a system disk for DECnet for OpenVMS (Phase IV) and another system disk for DECnet–Plus (Phase V). For more information, see the DECnet–Plus documentation.

### 4.2.2 Where to Install

You may want to set up common system disks according to these guidelines:

| IF you want the cluster to have... | THEN perform the installation or upgrade... |
|---|---|
| One common system disk for all computer members | Once on the cluster common system disk. |
| A combination of one or more common system disks and one or more local (individual) system disks | Either:<br>• Once for each system disk<br>or<br>• Once on a common system disk and then run the CLUSTER_CONFIG.COM procedure to create duplicate system disks (thus enabling systems to have their own local system disk) |

**Note:** If your cluster includes multiple common system disks, you must later coordinate system files to define the cluster operating environment, as described in Chapter 5.

**Reference:** See Section 8.5 for information about creating a duplicate system disk.

**Example:** If your OpenVMS Cluster consists of 10 computers, 4 of which boot from a common Alpha system disk, 2 of which boot from a second common Alpha system disk, 2 of which boot from a common VAX system disk, and 2 of which boot from their own local system disk, you need to perform an installation five times.

### 4.2.3 Information Required

Table 4–1 table lists the questions that the OpenVMS operating system installation procedure prompts you with and describes how certain system parameters are affected by responses you provide. You will notice that two of the prompts vary, depending on whether the node is running DECnet. The table also provides an example of an installation procedure that is taking place on a node named JUPITR.

**Important:** Be sure you determine answers to the questions before you begin the installation.

**Note about versions:** Refer to the appropriate OpenVMS *OpenVMS Release Notes* document for the required version numbers of hardware and firmware. When mixing versions of the operating system in an OpenVMS Cluster, check the release notes for information about compatibility.

**Reference:** Refer to the appropriate OpenVMS upgrade and installation manual for complete installation instructions.

**Table 4–1 Information Required to Perform an Installation**

<table>
<tr><th>Prompt</th><th>Response</th><th>Parameter</th></tr>
<tr><td>Will this node be a cluster member (Y/N)?</td><td>
<table>
<tr><th>WHEN you respond...</th><th>AND...</th><th>THEN the VAXcluster parameter is set to...</th></tr>
<tr><td>N</td><td>CI and DSSI hardware is not present</td><td>0 — Node will not participate in the OpenVMS Cluster.</td></tr>
<tr><td>N</td><td>CI and DSSI hardware is present</td><td>1 — Node will automatically participate in the OpenVMS Cluster in the presence of CI or DSSI hardware.</td></tr>
<tr><td>Y</td><td></td><td>2 — Node will participate in the OpenVMS Cluster.</td></tr>
</table>
</td><td>VAXCLUSTER</td></tr>
<tr><td>What is the node's DECnet node name?</td><td>If the node is running DECnet, this prompt, the following prompt, and the SCSSYSTEMID prompt are displayed. Enter the DECnet node name or the DECnet–Plus node synonym (for example, JUPITR). If a node synonym is not defined, SCSNODE can be any name from 1 to 6 alphanumeric characters in length. The name cannot include dollar signs ( $ ) or underscores ( _ ).</td><td>SCSNODE</td></tr>
<tr><td>What is the node's DECnet node address?</td><td>Enter the DECnet node address (for example, a valid address might be 2.211). If an address has not been assigned, enter 0 now and enter a valid address when you start DECnet (discussed later in this chapter).<br><br>For DECnet–Plus, this question is asked when nodes are configured with a Phase IV compatible address. If a Phase IV compatible address is not configured, then the SCSSYSTEMID system parameter can be set to any value.</td><td>SCSSYSTEMID</td></tr>
<tr><td>What is the node's SCS node name?</td><td>If the node is not running DECnet, this prompt and the following prompt are displayed in place of the two previous prompts. Enter a name of 1 to 6 alphanumeric characters that uniquely names this node. At least 1 character must be a letter. The name cannot include dollar signs ( $ ) or underscores ( _ ).</td><td>SCSNODE</td></tr>
<tr><td>What is the node's SCSSYSTEMID number?</td><td>This number must be unique within this cluster. SCSSYSTEMID is the low-order 32 bits of the 48-bit system identification number.<br><br>If the node is running DECnet for OpenVMS, calculate the value from the DECnet address using the following formula:<br><br>SCSSYSTEMID = (DECnet-area-number * 1024) + (DECnet-node-number)<br><br>Example: If the DECnet address is 2.211, calculate the value as follows:<br><br><code>SCSSYSTEMID = (2 * 1024) + 211 = 2259</code></td><td>SCSSYSTEMID</td></tr>
</table>

(continued on next page)

**Table 4–1 (Cont.) Information Required to Perform an Installation**

| Prompt | Response | | Parameter |
|---|---|---|---|
| Will the Ethernet be used for cluster communications (Y/N)?[1] | | | NISCS_LOAD_PEA0 |
| | **IF you respond...** | **THEN the NISCS_LOAD_PEA0 parameter is set to...** | |
| | N | 0 — PEDRIVER is not loaded[2]; cluster communications does not use Ethernet or FDDI. | |
| | Y | 1 — Loads PEDRIVER to enable cluster communications over Ethernet or FDDI. | |
| Enter this cluster's group number: | Enter a number in the range of 1 to 4095 or 61440 to 65535 (see Section 2.5). This value is stored in the CLUSTER_AUTHORIZE.DAT file in the SYS$COMMON:[SYSEXE] directory. | | Not applicable |
| Enter this cluster's password: | Enter the cluster password. The password must be from 1 to 31 alphanumeric characters in length and can include dollar signs ( $ ) and underscores ( _ ) (see Section 2.5). This value is stored in scrambled form in the CLUSTER_AUTHORIZE.DAT file in the SYS$COMMON:[SYSEXE] directory. | | Not applicable |
| Reenter this cluster's password for verification: | Reenter the password. | | Not applicable |
| Will JUPITR be a disk server (Y/N)? | | | MSCP_LOAD |
| | **IF you respond...** | **THEN the MSCP_LOAD parameter is set to...** | |
| | N | 0 — The MSCP server will not be loaded. This is the correct setting for configurations in which all OpenVMS Cluster nodes can directly access all shared storage and do not require LAN failover. | |
| | Y | 1 — Loads the MSCP server with attributes specified by the MSCP_SERVE_ALL parameter, using the default CPU load capacity. | |
| Will JUPITR serve HSC or RF disks (Y/N)? | | | MSCP_SERVE_ALL |
| | **IF you respond...** | **THEN the MSCP_SERVE_ALL parameter is set to...** | |
| | Y | 1 — Serves all available disks. | |
| | N | 2 — Serves only locally connected (not HSC, HSJ, or RF) disks. | |

[1]All references to the Ethernet are also applicable to FDDI.

[2]PEDRIVER is the LAN port emulator driver that implements the NISCA protocol and controls communications between local and remote LAN ports.

(continued on next page)

**Table 4–1 (Cont.) Information Required to Perform an Installation**

| Prompt | Response | Parameter |
|---|---|---|
| Enter a value for JUPITR's ALLOCLASS parameter:[3] | The value is dependent on the system configuration:<br>• If the system will serve RF disks, assign a nonzero value to the allocation class.<br>**Reference:** See Section 6.2.2.5 to assign DSSI allocation classes.<br>• If the system will serve HSC disks, enter the allocation class value of the HSC.<br>**Reference:** See Section 6.2.2.2 to assign HSC allocation classes.<br>• If the system will serve HSJ disks, enter the allocation class value of the HSJ.<br>**Reference:** For complete information about the HSJ console commands, refer to the HSJ hardware documentation. See Section 6.2.2.3 to assign HSJ allocation classes.<br>• If the system will serve HSD disks, enter the allocation class value of the HSD.<br>**Reference:** See Section 6.2.2.4 to assign HSC allocation classes.<br>• If the system disk is connected to a dual-pathed disk, enter a value from 1 to 255 that will be used on both storage controllers.<br>• If the system is connected to a shared SCSI bus (it shares storage on that bus with another system) and if it does not use port allocation classes for naming the SCSI disks, enter a value from 1 to 255. This value must be used by all the systems and disks connected to the SCSI bus.<br>**Reference:** For complete information about port allocation classes, see Section 6.2.1.<br>• If the system will use Volume Shadowing for OpenVMS, enter a value from 1 to 255.<br>**Reference:** For more information, see *Volume Shadowing for OpenVMS*.<br>• If none of the above are true, enter 0 (zero). | ALLOCLASS |
| Does this cluster contain a quorum disk [N]? | Enter Y or N, depending on your configuration. If you enter Y, the procedure prompts for the name of the quorum disk. Enter the device name of the quorum disk. (Quorum disks are discussed in Chapter 2.) | DISK_ QUORUM |

[3]Refer to Section 6.2 for complete information about device naming conventions.

## 4.3 Installing Software Licenses

While rebooting at the end of the installation procedure, the system displays messages warning that you must install the operating system software and the OpenVMS Cluster software license. The OpenVMS Cluster software supports the OpenVMS License Management Facility (LMF). License units for clustered systems are allocated on an unlimited system-use basis.

### 4.3.1 Guidelines

Be sure to install all OpenVMS Cluster licenses and all licenses for layered products and DECnet as soon as the system is available. Procedures for installing licenses are described in the release notes distributed with the software kit and in the *OpenVMS License Management Utility Manual*. Additional licensing information is described in the respective SPDs.

Use the following guidelines when you install software licenses:

- Install an OpenVMS Cluster Software for Alpha license for each Alpha processor in the OpenVMS Cluster.
- Install an OpenVMS Cluster Software for VAX license for each VAX processor in an OpenVMS Cluster system.
- Install or upgrade licenses for layered products that will run on all nodes in an OpenVMS Cluster system.
- OpenVMS Product Authorization Keys (PAKs) that have the Alpha option can be loaded and used only on Alpha processors. However, PAKs can be located in a **license database (LDB)** that is shared by both Alpha and VAX processors.
- Do not load Availability PAKs for VAX systems (Availability PAKs that do not include the Alpha option) on Alpha systems.
- PAK types such as Activity PAKs (also known as concurrent or n-user PAKs) and Personal Use PAKs (identified by the RESERVE_UNITS option) work on both VAX and Alpha systems.
- Compaq recommends that you perform licensing tasks using an Alpha LMF.

## 4.4 Installing Layered Products

By installing layered products before other nodes are added to the OpenVMS Cluster, the software is installed automatically on new members when they are added to the OpenVMS Cluster system.

**Note:** For clusters with multiple system disks (VAX, Alpha, or both) you must perform a separate installation for each system disk.

### 4.4.1 Procedure

Table 4–2 describes the actions you take to install layered products on a common system disk.

**Table 4–2 Installing Layered Products on a Common System Disk**

| Phase | Action |
|---|---|
| Before installation | Perform one or more of the following steps, as necessary for your system.<br>1. Check each node's system parameters and modify the values, if necessary. Refer to the layered-product installation guide or release notes for information about adjusting system parameter values.<br>2. If necessary, disable logins on each node that boots from the disk using the DCL command SET LOGINS/INTERACTIVE=0. Send a broadcast message to notify users about the installation. |
| Installation | Refer to the appropriate layered-product documentation for product-specific installation information. Perform the installation once for each system disk. |
| After installation | Perform one or more of the following steps, as necessary for your system.<br>1. If necessary, create product-specific files in the SYS$SPECIFIC directory on each node. (The installation utility describes whether or not you need to create a directory in SYS$SPECIFIC.) When creating files and directories, be careful to specify exactly where you want the file to be located:<br>• Use SYS$SPECIFIC or SYS$COMMON instead of SYS$SYSROOT.<br>• Use SYS$SPECIFIC:[SYSEXE] or SYS$COMMON:[SYSEXE] instead of SYS$SYSTEM.<br>**Reference:** Section 5.3 describes directory structures in more detail.<br>2. Modify files in SYS$SPECIFIC if the installation procedure tells you to do so. Modify files on each node that boots from this system disk.<br>3. Reboot each node to ensure that:<br>• The node is set up to run the layered product correctly.<br>• The node is running the latest version of the layered product.<br>4. Manually run the installation verification procedure (IVP) if you did not run it during the layered product installation. Run the IVP from at least one node in the OpenVMS Cluster, but preferably from all nodes that boot from this system disk. |

## 4.5 Configuring and Starting a Satellite Booting Service

After you have installed the operating system and the required licenses on the first OpenVMS Cluster computer, you can configure and start a satellite booting service. You can use the LANCP utility, or DECnet software, or both.

Compaq recommends LANCP for booting OpenVMS Cluster satellites. LANCP has shipped with the OpenVMS operating system since Version 6.2. It provides a general-purpose MOP booting service that can be used for booting satellites into an OpenVMS Cluster. (LANCP can service all types of MOP downline load requests, including those from terminal servers, LAN resident printers, and X terminals, and can be used to customize your LAN environment.)

DECnet provides a MOP booting service for booting OpenVMS Cluster satellites, as well as other local and wide area network services, including task-to-task communications for applications.

**Note**

If you plan to use LANCP in place of DECnet, and you also plan to move from DECnet Phase IV to DECnet–Plus, Compaq recommends the following order:

1. Replace DECnet with LANCP for satellite booting (MOP downline load service) using LAN$POPULATE.COM.
2. Migrate from DECnet Phase IV to DECnet-Plus.

There are two cluster configuration command procedures, CLUSTER_CONFIG_LAN.COM and CLUSTER_CONFIG.COM. CLUSTER_CONFIG_LAN.COM uses LANCP to provide MOP services to boot satellites; CLUSTER_CONFIG.COM uses DECnet for the same purpose.

Before choosing LANCP, DECnet, or both, consider the following factors:

- Applications you will be running on your cluster

  DECnet task-to-task communications is a method commonly used for communication between programs that run on different nodes in a cluster or a network. If you are running a program with that dependency, you need to run DECnet. If you are not running any programs with that dependency, you do not need to run DECnet.

- Limiting applications that require DECnet to certain nodes in your cluster

  If you are running applications that require DECnet task-to-task communications, you can run those applications on a subset of the nodes in your cluster and restrict DECnet usage to those nodes. You can use LANCP software on the remaining nodes and use a different network, such as DIGITAL TCP/IP Services for OpenVMS, for other network services.

- Managing two types of software for the same purpose

  If you are already using DECnet for booting satellites, you may not want to introduce another type of software for that purpose. Introducing any new software requires time to learn and manage it.

- LANCP MOP services can coexist with DECnet MOP services in an OpenVMS Cluster in the following ways:

  – Running on different systems

    For example, DECnet MOP service is enabled on some of the systems on the LAN and LAN MOP is enabled on other systems.

  – Running on different LAN devices on the same system

    For example, DECnet MOP service is enabled on a subset of the available LAN devices on the system and LAN MOP is enabled on the remainder.

  – Running on the same LAN device on the same system but targeting a different set of nodes for service

    For example, both DECnet MOP and LAN MOP are enabled but LAN MOP has limited the nodes to which it will respond. This allows DECnet MOP to respond to the remaining nodes.

Instructions for configuring both LANCP and DECnet are provided in this section.

### 4.5.1 Configuring and Starting the LANCP Utility

You can use the LAN Control Program (LANCP) utility to configure a local area network (LAN). You can also use the LANCP utility, in place of DECnet or in addition to DECnet, to provide support for booting satellites in an OpenVMS Cluster and for servicing all types of MOP downline load requests, including those from terminal servers, LAN resident printers, and X terminals.

**Reference:** For more information about using the LANCP utility to configure a LAN, see the *OpenVMS System Manager's Manual, Volume 2: Tuning, Monitoring, and Complex Systems* and the *OpenVMS System Management Utilities Reference Manual: A–L*.

### 4.5.2 Booting Satellite Nodes with LANCP

The LANCP utility provides a general-purpose MOP booting service that can be used for booting satellites into an OpenVMS Cluster. It can also be used to service all types of MOP downline load requests, including those from terminal servers, LAN resident printers, and X terminals. To use LANCP for this purpose, all OpenVMS Cluster nodes must be running OpenVMS Version 6.2 or higher.

The CLUSTER_CONFIG_LAN.COM cluster configuration command procedure uses LANCP in place of DECnet to provide MOP services to boot satellites.

**Note:** If you plan to use LANCP in place of DECnet, and you also plan to move from DECnet for OpenVMS (Phase IV) to DECnet–Plus, Compaq recommends the following order:

1. Replace DECnet with LANCP for satellite booting (MOP downline load service), using LAN$POPULATE.COM.
2. Migrate from DECnet for OpenVMS to DECnet–Plus.

### 4.5.3 Data Files Used by LANCP

LANCP uses the following data files:

- SYS$SYSTEM:LAN$DEVICE_DATABASE.DAT

  This file maintains information about devices on the local node. By default, the file is created in SYS$SPECIFIC:[SYSEXE], and the system looks for the file in that location. However, you can modify the file name or location for this file by redefining the systemwide logical name LAN$DEVICE_DATABASE.

- SYS$SYSTEM:LAN$NODE_DATABASE.DAT

  This file contains information about the nodes for which LANCP will supply boot service. This file should be shared among all nodes in the OpenVMS Cluster, including both Alpha and VAX systems. By default, the file is created in SYS$COMMON:[SYSEXE], and the system looks for the file in that location. However, you can modify the file name or location for this file by redefining the systemwide logical name LAN$NODE_DATABASE.

### 4.5.4 Using LAN MOP Services in New Installations

To use LAN MOP services for satellite booting in new installations, follow these steps:

1. Add the startup command for LANCP.

You should start up LANCP as part of your system startup procedure. To do this, remove the comment from the line in SYS$MANAGER:SYSTARTUP_VMS.COM that runs the LAN$STARTUP command procedure. If your OpenVMS Cluster system will have more than one system disk, see Section 4.5.3 for a description of logicals that can be defined for locating LANCP configuration files.

```
$ @SYS$STARTUP:LAN$STARTUP
```

You should now either reboot the system or invoke the preceding command procedure from the system manager's account to start LANCP.

2. Follow the steps in Chapter 8 for configuring an OpenVMS Cluster system and adding satellites. Use the CLUSTER_CONFIG_LAN.COM command procedure instead of CLUSTER_CONFIG.COM. If you invoke CLUSTER_CONFIG.COM, it gives you the option to switch to running CLUSTER_CONFIG_LAN.COM if the LANCP process has been started.

### 4.5.5 Using LAN MOP Services in Existing Installations

To migrate from DECnet MOP services to LAN MOP services for satellite booting, follow these steps:

1. Redefine the LANCP database logical names.

   This step is optional. If you want to move the data files used by LANCP, LAN$DEVICE_DATABASE and LAN$NODE_DATABASE, off the system disk, redefine their systemwide logical names. Add the definitions to the system startup files.

2. Use LANCP to create the LAN$DEVICE_DATABASE

   The permanent LAN$DEVICE_DATABASE is created when you issue the first LANCP DEVICE command. To create the database and get a list of available devices, enter the following commands:

```
$ MCR LANCP
LANCP> LIST DEVICE /MOPDLL
%LANCP-I-FNFDEV, File not found, LAN$DEVICE_DATABASE
%LANACP-I-CREATDEV, Created LAN$DEVICE_DATABASE file

Device Listing, permanent database:
  --- MOP Downline Load Service Characteristics ---
Device    State   Access Mode      Client            Data Size
------    -----   -----------      ------            ---------
ESA0    Disabled NoExlusive  NoKnownClientsOnly     246 bytes
FCA0    Disabled NoExlusive  NoKnownClientsOnly     246 bytes
```

3. Use LANCP to enable LAN devices for MOP booting.

   By default, the LAN devices have MOP booting capability disabled. Determine the LAN devices for which you want to enable MOP booting. Then use the DEFINE command in the LANCP utility to enable these devices to service MOP boot requests in the permanent database, as shown in the following example:

```
LANCP> DEFINE DEVICE ESA0:/MOP=ENABLE
```

4. Run LAN$POPULATE.COM (found in SYS$EXAMPLES) to obtain MOP booting information and to produce LAN$DEFINE and LAN$DECNET_MOP_CLEANUP, which are site specific.

LAN$POPULATE extracts all MOP booting information from a DECnet Phase IV NETNODE_REMOTE.DAT file or from the output of the DECnet–Plus NCL command SHOW MOP CLIENT * ALL.

For DECnet Phase IV sites, the LAN$POPULATE procedure scans all DECnet areas (1–63) by default. If you MOP boot systems from only a single or a few DECnet areas, you can cause the LAN$POPULATE procedure to operate on a single area at a time by providing the area number as the P1 parameter to the procedure, as shown in the following example (including log):

```
$ @SYS$EXAMPLES:LAN$POPULATE 15

 LAN$POPULATE - V1.0

 Do you want help (Y/N) <N>:

 LAN$DEFINE.COM has been successfully created.

 To apply the node definitions to the LANCP permanent database,
 invoke the created LAN$DEFINE.COM command procedure.

        Compaq recommends that you review LAN$DEFINE.COM and remove any
        obsolete entries prior to executing this command procedure.

 A total of 2 MOP definitions were entered into LAN$DEFINE.COM
```

5. Run LAN$DEFINE.COM to populate LAN$NODE_DATABASE.

   LAN$DEFINE populates the LANCP downline loading information into the LAN node database, SYS$COMMON:[SYSEVE]LAN$NODE_DATABASE.DAT file. Compaq recommends that you review LAN$DEFINE.COM and remove any obsolete entries before executing it.

   In the following sequence, the LAN$DEFINE.COM procedure that was just created is displayed on the screen and then executed:

```
$ TYPE LAN$DEFINE.COM

 $ !
 $ ! This file was generated by LAN$POPULATE.COM on 16-DEC-1996 09:20:31
 $ ! on node CLU21.
 $ !
 $ ! Only DECnet Area 15 was scanned.
 $ !
 $ MCR LANCP
 Define Node PORK    /Address=08-00-2B-39-82-85 /File=APB.EXE -
                  /Root=$21$DKA300:<SYS11.> /Boot_type=Alpha_Satellite
 Define Node JYPIG   /Address=08-00-2B-A2-1F-81 /File=APB.EXE -
                  /Root=$21$DKA300:<SYS10.> /Boot_type=Alpha_Satellite
 EXIT

$ @LAN$DEFINE

 %LANCP-I-FNFNOD, File not found, LAN$NODE_DATABASE
 -LANCP-I-CREATNOD, Created LAN$NODE_DATABASE file
 $
```

The following example shows a LAN$DEFINE.COM command procedure that was generated by LAN$POPULATE for migration from DECnet–Plus to LANCP.

```
$ ! LAN$DEFINE.COM - LAN MOP Client Setup
$ !
$ ! This file was generated by LAN$POPULATE.COM at  8-DEC-1996 14:28:43.31
$ ! on node BIGBOX.
$ !
$ SET NOON
$ WRITE SYS$OUTPUT "Setting up MOP DLL clients in LANCP...
$ MCR LANCP
SET    NODE SLIDER
/ADDRESS=08-00-2B-12-D8-72/ROOT=BIGBOX$DKB0:<SYS10.>/BOOT_TYP
E=VAX_satellite/FILE=NISCS_LOAD.EXE
DEFINE NODE SLIDER
/ADDRESS=08-00-2B-12-D8-72/ROOT=BIGBOX$DKB0:<SYS10.>/BOOT_TYP
E=VAX_satellite/FILE=NISCS_LOAD.EXE
EXIT
$ !
$  WRITE SYS$OUTPUT "DECnet Phase V to LAN MOPDLL client migration complete!"
$  EXIT
```

6. Run LAN$DECNET_MOP_CLEANUP.COM.

   You can use LAN$DECNET_MOP_CLEANUP.COM to remove the clients' MOP downline loading information from the DECnet database. Compaq recommends that you review LAN$DECNET_MOP_CLEANUP.COM and remove any obsolete entries before executing it.

   The following example shows a LAN$DECNET_MOP_CLEANUP.COM command procedure that was generated by LAN$POPULATE for migration from DECnet–Plus to LANCP.

   **Note:** When migrating from DECnet–Plus, additional cleanup is necessary. You must edit your NCL scripts (*.NCL) manually.

```
$ ! LAN$DECNET_MOP_CLEANUP.COM - DECnet MOP Client Cleanup
$ !
$ ! This file was generated by LAN$POPULATE.COM at  8-DEC-1995 14:28:43.47
$ ! on node BIGBOX.
$ !
$ SET NOON
$ WRITE SYS$OUTPUT "Removing MOP DLL clients from DECnet database..."
$ MCR NCL
DELETE NODE 0 MOP CLIENT SLIDER
EXIT
$ !
$  WRITE SYS$OUTPUT "DECnet Phase V MOPDLL client cleanup complete!"
$  EXIT
```

7. Start LANCP.

   To start LANCP, execute the startup command procedure as follows:

```
$ @SYS$STARTUP:LAN$STARTUP
  %RUN-S-PROC_ID, identification of created process is 2920009B
  $
```

   You should start up LANCP for all boot nodes as part of your system startup procedure. To do this, include the following line in your site-specific startup file (SYS$MANAGER:SYSTARTUP_VMS.COM):

```
$ @SYS$STARTUP:LAN$STARTUP
```

   If you have defined logicals for either LAN$DEVICE_DATABASE or LAN$NODE_DATABASE, be sure that these are defined in your startup files prior to starting up LANCP.

8. Disable DECnet MOP booting.

   If you use LANCP for satellite booting, you may no longer need DECnet to handle MOP requests. If this is the case for your site, you can turn off this capability with the appropriate NCP command (DECnet for OpenVMS) or NCL commands (DECnet–Plus).

For more information about the LANCP utility, see the *OpenVMS System Manager's Manual* and the *OpenVMS System Management Utilities Reference Manual*.

### 4.5.6 Configuring DECnet

The process of configuring the DECnet network typically entails several operations, as shown in Table 4–3. An OpenVMS Cluster running both implementations of DECnet requires a system disk for DECnet for OpenVMS (Phase IV) and another system disk for DECnet–Plus (Phase V).

**Note:** DECnet for OpenVMS implements Phase IV of Digital Network Architecture (DNA). DECnet–Plus implements Phase V of DNA. The following discussions are specific to the DECnet for OpenVMS product.

**Reference:** Refer to the DECnet–Plus documentation for equivalent DECnet–Plus configuration information.

**Table 4–3 Procedure for Configuring the DECnet Network**

| Step | Action |
|---|---|
| **1** | Log in as system manager and execute the NETCONFIG.COM command procedure as shown. Enter information about your node when prompted. Note that DECnet–Plus nodes execute the NET$CONFIGURE.COM command procedure.<br><br>**Reference:** See the DECnet for OpenVMS or the DECnet–Plus documentation, as appropriate, for examples of these procedures. |
| **2** | When a node uses multiple LAN adapter connections to the same LAN and also uses DECnet for communications, you must *disable DECnet use* of all but one of the LAN devices.<br><br>To do this, remove all but one of the lines and circuits associated with the adapters connected to the same LAN or extended LAN from the DECnet configuration database after the NETCONFIG.COM procedure is run.<br><br>For example, issue the following commands to invoke NCP and disable DECnet use of the LAN device XQB0:<br><br>`$ RUN SYS$SYSTEM:NCP`<br>`NCP> PURGE CIRCUIT QNA-1 ALL`<br>`NCP> DEFINE CIRCUIT QNA-1 STA OFF`<br>`NCP> EXIT`<br><br>**References:**<br><br>See *Guidelines for OpenVMS Cluster Configurations* for more information about distributing connections to LAN segments in OpenVMS Cluster configurations.<br><br>See the DECnet–Plus documentation for information about removing routing circuits associated with all but one LAN adapter. (Note that the LAN adapter issue is not a problem if the DECnet–Plus node uses extended addressing and does not have any Phase IV compatible addressing in use on any of the routing circuits.) |

(continued on next page)

**Table 4–3 (Cont.) Procedure for Configuring the DECnet Network**

| Step | Action |
|---|---|
| **3** | Make remote node data available clusterwide. NETCONFIG.COM creates in the SYS$SPECIFIC:[SYSEXE] directory the permanent remote-node database file NETNODE_REMOTE.DAT, in which remote-node data is maintained. To make this data available throughout the OpenVMS Cluster, you move the file to the SYS$COMMON:[SYSEXE] directory.<br><br>**Example:** Enter the following commands to make DECnet information available clusterwide:<br><br>`$ RENAME SYS$SPECIFIC:[SYSEXE]NETNODE_REMOTE.DAT SYS$COMMON:[SYSEXE]NETNODE_REMOTE.DAT`<br><br>If your configuration includes multiple system disks, you can set up a common NETNODE_REMOTE.DAT file automatically by using the following command in SYLOGICALS.COM:<br><br>`$ DEFINE/SYSTEM/EXE NETNODE_REMOTE` *`ddcu:[directory]`*`NETNODE_REMOTE.DAT`<br><br>**Notes:** Compaq recommends that you set up a common NETOBJECT.DAT file clusterwide in the same manner.<br><br>DECdns is used by DECnet–Plus nodes to manage node data (the namespace). For DECnet–Plus, Session Control Applications replace objects. |
| **4** | Designate and enable router nodes to support the use of a cluster alias. At least one node participating in a cluster alias must be configured as a level 1 router.<br><br>†On VAX systems, you can designate a computer as a router node when you execute NETCONFIG.COM (as shown in step 1).<br><br>‡On Alpha systems, you might need to enable level 1 routing manually because the NETCONFIG.COM procedure does not prompt you with the routing question.<br><br>Depending on whether the configuration includes all Alpha nodes or a combination of VAX and Alpha nodes, follow these instructions: |

| IF the cluster consists of... | THEN... |
|---|---|
| Alpha nodes only | You must enable level 1 routing manually (see the example below) on one of the Alpha nodes. |
| Both Alpha and VAX nodes | You do not need to enable level 1 routing on an Alpha node if one of the VAX nodes is already a routing node.<br><br>You do not need to enable the DECnet extended function license DVNETEXT on an Alpha node if one of the VAX nodes is already a routing node. |

‡**Example:** On Alpha systems, if you need to enable level 1 routing on an Alpha node, invoke the NCP utility to do so. For example:

```
$ RUN SYS$SYSTEM:NCP
NCP>  DEFINE EXECUTOR TYPE ROUTING IV
```

‡On Alpha systems, level 1 routing is supported to enable cluster alias operations only.

†VAX specific
‡Alpha specific

(continued on next page)

**Table 4–3 (Cont.) Procedure for Configuring the DECnet Network**

| Step | Action |
|---|---|
| 5 | Optionally, define a cluster alias. If you want to define a cluster alias, invoke the NCP utility to do so. The information you specify using these commands is entered in the DECnet permanent executor database and takes effect when you start the network.<br><br>**Example:** The following NCP commands establish SOLAR as an alias:<br><br>`$ RUN SYS$SYSTEM:NCP`<br>`NCP> DEFINE NODE 2.1 NAME SOLAR`<br>`NCP> DEFINE EXECUTOR ALIAS NODE SOLAR`<br>`NCP> EXIT`<br>`$`<br><br>**Reference:** Section 4.5.8 describes the cluster alias. Section 4.5.9 describes how to enable alias operations for other computers. See the DECnet–Plus documentation for information about setting up a cluster alias on DECnet–Plus nodes.<br><br>**Note:** DECnet for OpenVMS nodes and DECnet–Plus nodes cannot share a cluster alias. |

## 4.5.7 Starting DECnet

If you are using DECnet–Plus, a separate step is not required to start the network. DECnet–Plus starts automatically on the next reboot after the node has been configured using the NET$CONFIGURE.COM procedure.

If you are using DECnet for OpenVMS, at the system prompt, enter the following command to start the network:

```
$ @SYS$MANAGER:STARTNET.COM
```

To ensure that the network is started each time an OpenVMS Cluster computer boots, add that command line to the appropriate startup command file or files. (Startup command files are discussed in Section 5.6.)

## 4.5.8 What is the Cluster Alias?

The cluster alias acts as a single network node identifier for an OpenVMS Cluster system. When enabled, the cluster alias makes all the OpenVMS Cluster nodes appear to be one node from the point of view of the rest of the network.

Computers in the cluster can use the alias for communications with other computers in a DECnet network. For example, networked applications that use the services of an OpenVMS Cluster should use an alias name. Doing so ensures that the remote access will be successful when at least one OpenVMS Cluster member is available to process the client program's requests.

**Rules:**

- DECnet for OpenVMS (Phase IV) allows a maximum of 64 OpenVMS Cluster computers to participate in a cluster alias. If your cluster includes more than 64 computers, you must determine which 64 should participate in the alias and then define the alias on those computers.

  At least one of the OpenVMS Cluster nodes that uses the alias node identifier must have level 1 routing enabled.

  - On Alpha nodes, routing between multiple circuits is not supported. However, routing is supported to allow cluster alias operations. Level 1 routing is supported only for enabling the use of a cluster alias. The DVNETEXT PAK must be used to enable this limited function.
  - On VAX nodes, full level 1 routing support is available.

  - On both Alpha and VAX systems, all cluster nodes sharing the same alias node address must be in the same area.

- DECnet–Plus allows a maximum of 96 OpenVMS Cluster computers to participate in the cluster alias.

  DECnet–Plus does not require that a cluster member be a routing node, but an adjacent Phase V router is required to use a cluster alias for DECnet–Plus systems.

- A single cluster alias can include nodes running either DECnet for OpenVMS or DECnet–Plus, but not both.

### 4.5.9 Enabling Alias Operations

If you have defined a cluster alias and have enabled routing as shown in Section 4.5.6, you can enable alias operations for other computers *after the computers are up and running in the cluster*. To enable such operations (that is, to allow a computer to accept incoming connect requests directed toward the alias), follow these steps:

1. Log in as system manager and invoke the SYSMAN utility. For example:

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN>
```

2. At the SYSMAN> prompt, enter the following commands:

```
SYSMAN> SET ENVIRONMENT/CLUSTER
%SYSMAN-I-ENV, current command environment:
        Clusterwide on local cluster
        Username SYSTEM  will be used on nonlocal nodes
SYSMAN> SET PROFILE/PRIVILEGES=(OPER,SYSPRV)
SYSMAN> DO MCR NCP SET EXECUTOR STATE OFF
%SYSMAN-I-OUTPUT, command execution on node X...
   .
   .
   .
SYSMAN> DO MCR NCP DEFINE EXECUTOR ALIAS INCOMING ENABLED
%SYSMAN-I-OUTPUT, command execution on node X...
   .
   .
   .
SYSMAN> DO @SYS$MANAGER:STARTNET.COM
%SYSMAN-I-OUTPUT, command execution on node X...
   .
   .
   .
```

**Note:** Compaq does not recommend enabling alias operations for satellite nodes.

**Reference:** For more details about DECnet for OpenVMS networking and cluster alias, see the *DECnet for OpenVMS Networking Manual* and *DECnet for OpenVMS Network Management Utilities*. For equivalent information about DECnet–Plus, see the DECnet–Plus documentation.

# 5
# Preparing a Shared Environment

In any OpenVMS Cluster environment, it is best to share resources as much as possible. Resource sharing facilitates workload balancing because work can be distributed across the cluster.

## 5.1 Shareable Resources

Most, but not all, resources can be shared across nodes in an OpenVMS Cluster. The following table describes resources that can be shared.

| Shareable Resources | Description |
|---|---|
| System disks | All members of the same architecture[1] can share a single system disk, each member can have its own system disk, or members can use a combination of both methods. |
| Data disks | All members can share any data disks. For local disks, access is limited to the local node unless you explicitly set up the disks to be cluster accessible by means of the MSCP server. |
| Tape drives | All members can share tape drives. (Note that this does not imply that all members can have simultaneous access.) For local tape drives, access is limited to the local node unless you explicitly set up the tapes to be cluster accessible by means of the TMSCP server. Only DSA tapes can be served to all OpenVMS Cluster members. |
| Batch and print queues | Users can submit batch jobs to any queue in the OpenVMS Cluster, regardless of the processor on which the job will actually execute. Generic queues can balance the load among the available processors. |
| Applications | Most applications work in an OpenVMS Cluster just as they do on a single system. Application designers can also create applications that run simultaneously on multiple OpenVMS Cluster nodes, which share data in a file. |
| User authorization files | All nodes can use either a common user authorization file (UAF) for the same access on all systems or multiple UAFs to enable node-specific quotas. If a common UAF is used, all user passwords, directories, limits, quotas, and privileges are the same on all systems. |

[1]Data on system disks can be shared between Alpha and VAX processors. However, VAX nodes cannot boot from an Alpha system disk, and Alpha nodes cannot boot from a VAX system disk.

### 5.1.1 Local Resources

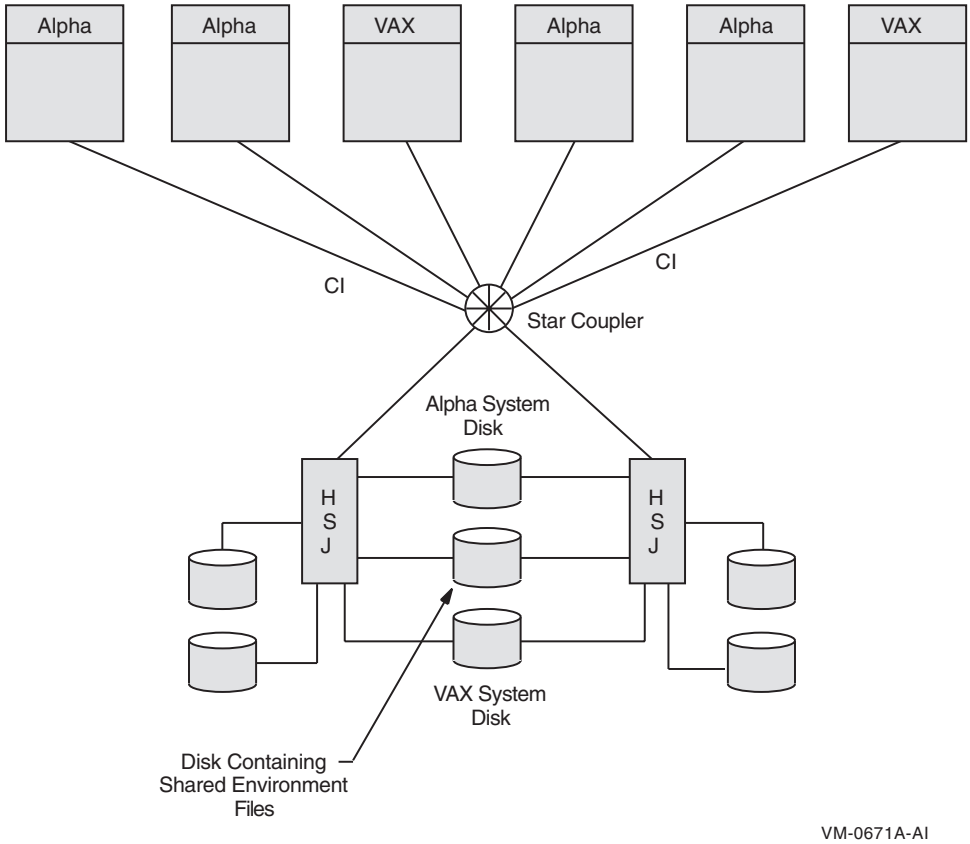
The following table lists resources that are accessible only to the local node.

| Nonshareable Resources | Description |
|---|---|
| Memory | Each OpenVMS Cluster member maintains its own memory. |
| User processes | When a user process is created on an OpenVMS Cluster member, the process must complete on that computer, using local memory. |
| Printers | A printer that does not accept input through queues is used only by the OpenVMS Cluster member to which it is attached. A printer that accepts input through queues is accessible by any OpenVMS Cluster member. |

### 5.1.2 Sample Configuration

Figure 5–1 shows an example of an OpenVMS Cluster system that has both an Alpha system disk and a VAX system disk, and a dual-ported disk that is set up so the environmental files can be shared between the Alpha and VAX systems.

**Figure 5–1 Resource Sharing in Mixed-Architecture Cluster System**




## 5.2 Common-Environment and Multiple-Environment Clusters

Depending on your processing needs, you can prepare either an environment in which all environmental files are shared clusterwide or an environment in which

some files are shared clusterwide while others are accessible only by certain computers.

The following table describes the characteristics of common- and multiple-environment clusters.

| Cluster Type | Characteristics | Advantages |
|---|---|---|
| **Common environment** | | |
| Operating environment is identical on all nodes in the OpenVMS Cluster. | The environment is set up so that:<br>• All nodes run the same programs, applications, and utilities.<br>• All users have the same type of user accounts, and the same logical names are defined.<br>• All users can have common access to storage devices and queues. (Note that access is subject to how access control list [ACL] protection is set up for each user.)<br>• All users can log in to any node in the configuration and work in the same environment as all other users. | Easier to manage because you use a common version of each system file. |
| **Multiple environment** | | |
| Operating environment can vary from node to node. | An individual processor or a subset of processors are set up to:<br>• Provide multiple access according to the type of tasks users perform and the resources they use.<br>• Share a set of resources that are not available on other nodes.<br>• Perform specialized functions using restricted resources while other processors perform general timesharing work.<br>• Allow users to work in environments that are specific to the node where they are logged in. | Effective when you want to share *some* data among computers but you also want certain computers to serve specialized needs. |

## 5.3 Directory Structure on Common System Disks

The installation or upgrade procedure for your operating system generates a **common system disk**, on which most operating system and optional product files are stored in a common root directory.

### 5.3.1 Directory Roots

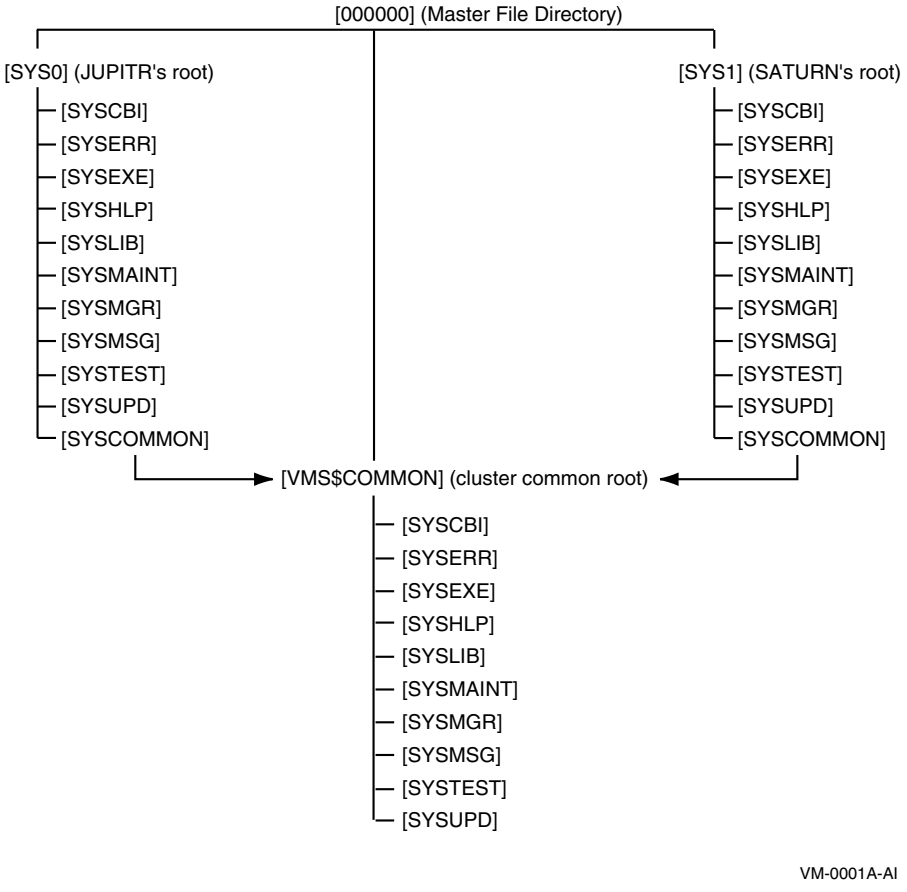
The system disk directory structure is the same on both Alpha and VAX systems. Whether the system disk is for Alpha or VAX, the entire directory structure—that is, the **common root** plus each computer's **local root**—is stored on the same disk. After the installation or upgrade completes, you use the CLUSTER_CONFIG.COM command procedure described in Chapter 8 to create a local root for each new computer to use when booting into the cluster.

In addition to the usual system directories, each local root contains a [SYS*n*.SYSCOMMON] directory that is a directory alias for [VMS$COMMON], the cluster common root directory in which cluster common files actually reside. When you add a computer to the cluster, CLUSTER_CONFIG.COM defines the common root directory alias.

### 5.3.2 Directory Structure Example

Figure 5–2 illustrates the directory structure set up for computers JUPITR and SATURN, which are run from a common system disk. The disk's master file directory (MFD) contains the local roots (SYS0 for JUPITR, SYS1 for SATURN) and the cluster common root directory, [VMS$COMMON].

**Figure 5–2 Directory Structure on a Common System Disk**

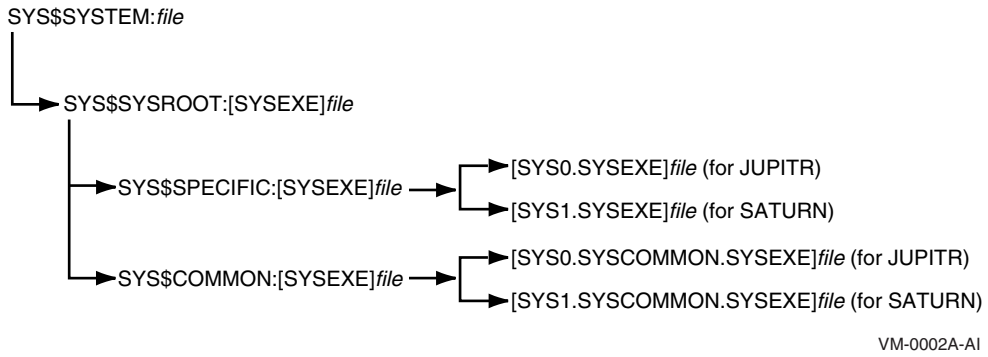


### 5.3.3 Search Order

The logical name SYS$SYSROOT is defined as a search list that points first to a local root (SYS$SYSDEVICE:[SYS0.SYSEXE]) and then to the common root (SYS$COMMON:[SYSEXE]). Thus, the logical names for the system directories (SYS$SYSTEM, SYS$LIBRARY, SYS$MANAGER, and so forth) point to two directories.

Figure 5–3 shows how directories on a common system disk are searched when the logical name SYS$SYSTEM is used in file specifications.

**Figure 5–3 File Search Order on Common System Disk**




**Important:** Keep this search order in mind when you manipulate system files on a common system disk. Computer-specific files must always reside and be updated in the appropriate computer's system subdirectory.

**Examples**

1. MODPARAMS.DAT must reside in SYS$SPECIFIC:[SYSEXE], which is [SYS0.SYSEXE] on JUPITR, and in [SYS1.SYSEXE] on SATURN. Thus, to create a new MODPARAMS.DAT file for JUPITR when logged in on JUPITR, enter the following command:

   ```
   $ EDIT SYS$SPECIFIC:[SYSEXE]MODPARAMS.DAT
   ```

   Once the file is created, you can use the following command to modify it when logged on to JUPITR:

   ```
   $ EDIT SYS$SYSTEM:MODPARAMS.DAT
   ```

   Note that if a MODPARAMS.DAT file does not exist in JUPITR's SYS$SPECIFIC:[SYSEXE] directory when you enter this command, but there is a MODPARAMS.DAT file in the directory SYS$COMMON:[SYSEXE], the command edits the MODPARAMS.DAT file in the common directory. If there is no MODPARAMS.DAT file in either directory, the command creates the file in JUPITR's SYS$SPECIFIC:[SYSEXE] directory.

2. To modify JUPITR's MODPARAMS.DAT when logged in on any other computer that boots from the same common system disk, enter the following command:

   ```
   $ EDIT SYS$SYSDEVICE:[SYS0.SYSEXE]MODPARAMS.DAT
   ```

3. To modify records in the cluster common system authorization file in a cluster with a single, cluster-common system disk, enter the following commands on any computer:

```
$ SET DEFAULT SYS$COMMON:[SYSEXE]
$ RUN SYS$SYSTEM:AUTHORIZE
```

4. To modify records in a computer-specific system authorization file when logged in to another computer that boots from the same cluster common system disk, you must set your default directory to the specific computer. For example, if you have set up a computer-specific system authorization file (SYSUAF.DAT) for computer JUPITR, you must set your default directory to JUPITR's computer-specific [SYSEXE] directory before invoking AUTHORIZE, as follows:

```
$ SET DEFAULT SYS$SYSDEVICE:[SYS0.SYSEXE]
$ RUN SYS$SYSTEM:AUTHORIZE
```

## 5.4 Clusterwide Logical Names

Clusterwide logical names were introduced in OpenVMS Version 7.2 for both OpenVMS Alpha and OpenVMS VAX. Clusterwide logical names extend the convenience and ease-of-use features of shareable logical names to OpenVMS Cluster systems.

Existing applications can take advantage of clusterwide logical names without any changes to the application code. Only a minor modification to the logical name tables referenced by the application (directly or indirectly) is required.

New logical names are local by default. Clusterwide is an attribute of a logical name table. In order for a new logical name to be clusterwide, it must be created in a clusterwide logical name table.

Some of the most important features of clusterwide logical names are:

- When a new node joins the cluster, it automatically receives the current set of clusterwide logical names.
- When a clusterwide logical name or name table is created, modified, or deleted, the change is automatically propagated to every other node in the cluster running OpenVMS Version 7.2 or later. Modifications include security profile changes to a clusterwide table.
- Translations are done locally so there is minimal performance degradation for clusterwide name translations.
- Because LNM$CLUSTER_TABLE and LNM$SYSCLUSTER_TABLE exist on all systems running OpenVMS Version 7.2 or later, the programs and command procedures that use clusterwide logical names can be developed, tested, and run on nonclustered systems.

### 5.4.1 Default Clusterwide Logical Name Tables

To support clusterwide logical names, the operating system creates two clusterwide logical name tables and their logical names at system startup, as shown in Table 5–1. These logical name tables and logical names are in addition to the ones supplied for the process, job, group, and system logical name tables. The names of the clusterwide logical name tables are contained in the system logical name directory, LNM$SYSTEM_DIRECTORY.

**Table 5–1 Default Clusterwide Logical Name Tables and Logical Names**

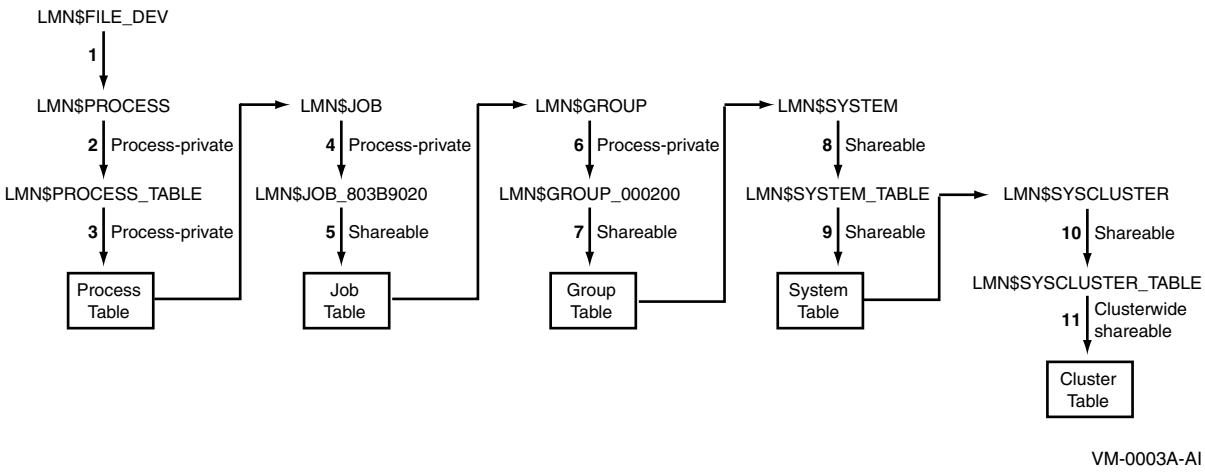
| Name | Purpose |
|---|---|
| LNM$SYSCLUSTER_TABLE | The default table for clusterwide system logical names. It is empty when shipped. This table is provided for system managers who want to use clusterwide logical names to customize their environments. The names in this table are available to anyone translating a logical name using SHOW LOGICAL/SYSTEM, specifying a table name of LNM$SYSTEM, or LNM$DCL_LOGICAL (DCL's default table search list), or LNM$FILE_DEV (system and RMS default). |
| LNM$SYSCLUSTER | The logical name for LNM$SYSCLUSTER_TABLE. It is provided for convenience in referencing LNM$SYSCLUSTER_TABLE. It is consistent in format with LNM$SYSTEM_TABLE and its logical name, LNM$SYSTEM. |
| LNM$CLUSTER_TABLE | The parent table for all clusterwide logical name tables, including LNM$SYSCLUSTER_TABLE. When you create a new table using LNM$CLUSTER_TABLE as the parent table, the new table will be available clusterwide. |
| LNM$CLUSTER | The logical name for LNM$CLUSTER_TABLE. It is provided for convenience in referencing LNM$CLUSTER_TABLE. |

### 5.4.2 Translation Order

The definition of LNM$SYSTEM has been expanded to include LNM$SYSCLUSTER. When a system logical name is translated, the search order is LNM$SYSTEM_TABLE, LNM$SYSCLUSTER_TABLE. Because the definitions for the system default table names, LNM$FILE_DEV and LNM$DCL_LOGICALS, include LNM$SYSTEM, translations using those default tables include definitions in LNM$SYSCLUSTER.

The current precedence order for resolving logical names is preserved. Clusterwide logical names that are translated against LNM$FILE_DEV are resolved last, after system logical names. The precedence order, from first to last, is process –> job –> group –> system –> cluster, as shown in Figure 5–4.

**Figure 5–4 Translation Order Specified by LNM$FILE_DEV**




## 5.4.3 Creating Clusterwide Logical Name Tables

You might want to create additional clusterwide logical name tables for the following purposes:

- For a multiprocess clusterwide application to use
- For members of a UIC group to share

To create a clusterwide logical name table, you must have create (C) access to the parent table and write (W) access to LNM$SYSTEM_DIRECTORY, or the SYSPRV (system) privilege.

A shareable logical name table has UIC-based protection. Each class of user (system (S), owner (O), group (G), and world (W)) can be granted four types of access: read (R), write (W), create (C), or delete (D).

You can create additional clusterwide logical name tables in the same way that you can create additional process, job, and group logical name tables—with the CREATE/NAME_TABLE command or with the $CRELNT system service. When creating a clusterwide logical name table, you must specify the /PARENT_TABLE qualifier and provide a value for the qualifier that is a clusterwide table name. Any existing clusterwide table used as the parent table will make the new table clusterwide.

The following example shows how to create a clusterwide logical name table:

```
$ CREATE/NAME_TABLE/PARENT_TABLE=LNM$CLUSTER_TABLE -
_$ new-clusterwide-logical-name-table
```

## 5.4.4 Alias Collisions Involving Clusterwide Logical Name Tables

Alias collisions involving clusterwide logical name tables are treated differently from alias collisions of other types of logical name tables. Table 5–2 describes the types of collisions and their outcomes.

**Table 5–2 Alias Collisions and Outcomes**

| Collision Type | Outcome |
|---|---|
| Creating a local table with same name and access mode as an existing clusterwide table | New local table is not created. The condition value SS$_NORMAL is returned, which means that the service completed successfully but the logical name table already exists. The existing clusterwide table and its names on all nodes remain in effect. |
| Creating a clusterwide table with same name and access mode as an existing local table | New clusterwide table is created. The condition value SS$_LNMCREATED is returned, which means that the logical name table was created. The local table and its names are deleted. If the clusterwide table was created with the DCL command DEFINE, a message is displayed:<br>DCL-I-TABSUPER, previous table *table_name* has been superseded<br>If the clusterwide table was created with the $CRELNT system service, $CRELNT returns the condition value: SS$_SUPERSEDE. |
| Creating a clusterwide table with same name and access mode as an existing clusterwide table | New clusterwide table is not created. The condition value SS$_NORMAL is returned, which means that the service completed successfully but the logical name table already exists. The existing table and all its names remain in effect, regardless of the setting of the $CRELNT system service's CREATE-IF attribute. This prevents surprise implicit deletions of existing table names from other nodes. |

### 5.4.5 Creating Clusterwide Logical Names

To create a clusterwide logical name, you must have write (W) access to the table in which the logical name is to be entered, or SYSNAM privilege if you are creating clusterwide logical names only in LNM$SYSCLUSTER. Unless you specify an access mode (user, supervisor, and so on), the access mode of the logical name you create defaults to the access mode from which the name was created. If you created the name with a DCL command, the access mode defaults to supervisor mode. If you created the name with a program, the access mode typically defaults to user mode.

When you create a clusterwide logical name, you must include the name of a clusterwide logical name table in the definition of the logical name. You can create clusterwide logical names by using DCL commands or with the $CRELNM system service.

The following example shows how to create a clusterwide logical name in the default clusterwide logical name table, LNM$CLUSTER_TABLE, using the DEFINE command:

```
$ DEFINE/TABLE=LNM$CLUSTER_TABLE logical-name equivalence-string
```

To create clusterwide logical names that will reside in a clusterwide logical name table you created, you define the new clusterwide logical name with the DEFINE command, specifying your new clusterwide table's name with the /TABLE qualifier, as shown in the following example:

```
$ DEFINE/TABLE=new-clusterwide-logical-name-table logical-name -
_$ equivalence-string
```

**Note**

If you attempt to create a new clusterwide logical name with the same access mode and identical equivalence names and attributes as an existing clusterwide logical name, the existing name is *not* deleted, and no messages are sent to remote nodes. This behavior differs from similar attempts for other types of logical names, which delete the existing name and create the new one. For clusterwide logical names, this difference is a performance enhancement.

The condition value SS$_NORMAL is returned. The service completed successfully, but the new logical name was not created.

### 5.4.6 Management Guidelines

When using clusterwide logical names, observe the following guidelines:

1. Do not use certain logical names clusterwide.

   The following logical names are not valid for clusterwide use:

   - Mailbox names, because mailbox devices are local to a node.
   - SYS$NODE and SYS$NODE_FULLNAME must be in LNM$SYSTEM_TABLE and are node specific.
   - LMF$LICENSE_TABLE.

2. Do not redefine LNM$SYSTEM.

   LNM$SYSTEM is now defined as LNM$SYSTEM_TABLE, LNM$SYSCLUSTER_TABLE. Do not reverse the order of these two tables. If you do, then any names created using the /SYSTEM qualifier or in LNM$SYSTEM would go in LNM$SYSCLUSTER_TABLE and be clusterwide. Various system failures would result. For example, the MOUNT/SYSTEM command would attempt to create a clusterwide logical name for a mounted volume, which would result in an error.

3. Keep LNM$SYSTEM contents in LNM$SYSTEM.

   Do not merge the logical names in LNM$SYSTEM into LNM$SYSCLUSTER. Many system logical names in LNM$SYSTEM contain system roots and either node-specific devices, or node-specific directories, or both.

4. Adopt naming conventions for logical names used at your site.

   To avoid confusion and name conflicts, develop one naming convention for system-specific logical names and another for clusterwide logical names.

5. Avoid using the dollar sign ($) in your own site's logical names, because OpenVMS software uses it in its names.

6. Be aware that clusterwide logical name operations will stall when the clusterwide logical name database is not consistent.

   This can occur during system initialization when the system's clusterwide logical name database is not completely initialized. It can also occur when the cluster server process has not finished updating the clusterwide logical name database, or during resynchronization after nodes enter or leave the cluster.

As soon as consistency is reestablished, the processing of clusterwide logical name operations resumes.

## 5.4.7 Using Clusterwide Logical Names in Applications

The $TRNLNM system service and the $GETSYI system service provide attributes that are specific to clusterwide logical names. This section describes those attributes. It also describes the use of $CRELNT as it pertains to creating a clusterwide table. For more information about using logical names in applications, refer to the *OpenVMS Programming Concepts Manual*.

### 5.4.7.1 Clusterwide Attributes for $TRNLNM System Service

Two clusterwide attributes are available in the $TRNLNM system service:

- LNM$V_CLUSTERWIDE
- LNM$M_INTERLOCKED

LNM$V_CLUSTERWIDE is an output attribute to be returned in the itemlist if you asked for the LNM$_ATTRIBUTES item for a logical name that is clusterwide.

LNM$M_INTERLOCKED is an **attr** argument bit that can be set to ensure that any clusterwide logical name modifications in progress are completed before the name is translated. LNM$M_INTERLOCKED is not set by default. If your application requires translation using the most recent definition of a clusterwide logical name, use this attribute to ensure that the translation is stalled until all pending modifications have been made.

On a single system, when one process modifies the shareable part of the logical name database, the change is visible immediately to other processes on that node. Moreover, while the modification is in progress, no other process can translate or modify shareable logical names.

In contrast, when one process modifies the clusterwide logical name database, the change is visible immediately on that node, but it takes a short time for the change to be propagated to other nodes. By default, translations of clusterwide logical names are not stalled. Therefore, it is possible for processes on different nodes to translate a logical name and get different equivalence names when modifications are in progress.

The use of LNM$M_INTERLOCKED guarantees that your application will receive the most recent definition of a clusterwide logical name.

### 5.4.7.2 Clusterwide Attribute for $GETSYI System Service

The clusterwide attribute, SYI$_CWLOGICALS, has been added to the $GETSYI system service. When you specify SYI$_CWLOGICALS, $GETSYI returns the value 1 if the clusterwide logical name database has been initialized on the CPU, or the value 0 if it has not been initialized. Because this number is a Boolean value (1 or 0), the buffer length field in the item descriptor should specify 1 (byte). On a nonclustered system, the value of SYI$_CWLOGICALS is always 0.

### 5.4.7.3 Creating Clusterwide Tables with the $CRELNT System Service

When creating a clusterwide table, the $CRELNT requester must supply a table name. OpenVMS does not supply a default name for clusterwide tables because the use of default names enables a process without the SYSPRV privilege to create a shareable table.

## 5.5 Defining and Accessing Clusterwide Logical Names

Initializing the clusterwide logical name database on a booting node requires sending a message to another node and having its CLUSTER_SERVER process reply with one or messages containing a description of the database. The CLUSTER_SERVER process on the booting node requests system services to create the equivalent names and tables. How long this initialization takes varies with conditions such as the size of the clusterwide logical name database, the speed of the cluster interconnect, and the responsiveness of the CLUSTER_SERVER process on the responding node.

Until a booting node's copy of the clusterwide logical name database is consistent with the logical name databases of the rest of the cluster, any attempt on the booting node to create or delete clusterwide names or tables is stalled transparently. Because translations are not stalled by default, any attempt to translate a clusterwide name before the database is consistent may fail or succeed, depending on timing. To stall a translation until the database is consistent, specify the F$TRNLNM CASE argument as INTERLOCKED.

### 5.5.1 Defining Clusterwide Logical Names in SYSTARTUP_VMS.COM

In general, system managers edit the SYLOGICALS.COM command procedure to define site-specific logical names that take effect at system startup. However, Compaq recommends that, if possible, clusterwide logical names be defined in the SYSTARTUP_VMS.COM command procedure instead with the exception of those logical names discussed in Section 5.5.2. The reason for defining clusterwide logical names in SYSTARTUP_VMS.COM is that SYSTARTUP_VMS.COM is run at a much later stage in the booting process than SYLOGICALS.COM.

OpenVMS startup is single streamed and synchronous except for actions taken by created processes, such as the CLUSTER_SERVER process. Although the CLUSTER_SERVER process is created very early in startup, it is possible that when SYLOGICALS.COM is executed, the booting node's copy of the clusterwide logical name database has not been fully initialized. In such a case, a clusterwide definition in SYLOGICALS.COM would stall startup and increase the time it takes for the system to become operational.

OpenVMS will ensure that the clusterwide database has been initialized before SYSTARTUP_VMS.COM is executed.

### 5.5.2 Defining Certain Logical Names in SYLOGICALS.COM

To be effective, certain logical names, such as LMF$LICENSE, NET$PROXY, and VMS$OBJECTS must be defined earlier in startup than when SYSTARTUP_VMS.COM is invoked. Most such names are defined in SYLOGICALS.COM, with the exception of VMS$OBJECTS, which is defined in SYSECURITY.COM, and any names defined in SYCONFIG.COM.

Although Compaq recommends defining clusterwide logical names in SYSTARTUP_VMS.COM, to define these names to be clusterwide, you must do so in SYLOGICALS.COM or SYSECURITY.COM. Note that doing this may increase startup time.

Alternatively, you can take the traditional approach and define these names as systemwide logical names with the same definition on every node.

### 5.5.3 Using Conditional Definitions for Startup Command Procedures

For clusterwide definitions in any startup command procedure that is common to all cluster nodes, Compaq recommends that you use a conditional definition. For example:

```
$ IF F$TRNLNM("CLUSTER_APPS") .EQS. "" THEN -
_$ DEFINE/TABLE=LNM$SYSCLUSTER/EXEC CLUSTER_APPS -
_$ $1$DKA500:[COMMON_APPS]
```

A conditional definition can prevent unpleasant surprises. For example, suppose a system manager redefines a name that is also defined in SYSTARTUP_VMS.COM but does not edit SYSTARTUP_VMS.COM because the new definition is temporary. If a new node joins the cluster, the new node would initially receive the new definition. However, when the new node executes SYSTARTUP_VMS.COM, it will cause all the nodes in the cluster, including itself, to revert to the original value.

If you include a conditional definition in SYLOGICALS.COM or SYSECURITY.COM, specify the F$TRNLNM CASE argument as INTERLOCKED to ensure that clusterwide logical names have been fully initialized before the translation completes. An example of a conditional definition with the argument specified follows:

```
$ IF F$TRNLNM("CLUSTER_APPS",,,,"INTERLOCKED") .EQS. "" THEN -
_$ DEFINE/TABLE=LNM$SYSCLUSTER/EXEC CLUSTER_APPS -
_$ $1$DKA500:[COMMON_APPS]
```

**Note**

F$GETSYI ("CWLOGICALS") always returns a value of FALSE on a noncluster system. Procedures that are designed to run in both clustered and nonclustered environments should first determine whether they are in a cluster and, if so, then determine whether clusterwide logical names are initialized.

## 5.6 Coordinating Startup Command Procedures

Immediately after a computer boots, it runs the site-independent command procedure SYS$SYSTEM:STARTUP.COM to start up the system and control the sequence of startup events. The STARTUP.COM procedure calls a number of other startup command procedures that perform cluster-specific and node-specific tasks.

The following sections describe how, by setting up appropriate cluster-specific startup command procedures and other system files, you can prepare the OpenVMS Cluster operating environment on the first installed computer before adding other computers to the cluster.

**Reference:** See also the *OpenVMS System Manager's Manual* for more information about startup command procedures.

### 5.6.1 OpenVMS Startup Procedures

Several startup command procedures are distributed as part of the OpenVMS operating system. The SYS$SYSTEM:STARTUP.COM command procedure executes immediately after OpenVMS is booted and invokes the site-specific startup command procedures described in the following table.

| Procedure Name | Invoked by | Function |
|---|---|---|
| SYS$MANAGER: SYPAGSWPFILES.COM | SYS$SYSTEM: STARTUP.COM | A file to which you add commands to install page and swap files (other than the primary page and swap files that are installed automatically). |
| SYS$MANAGER: SYCONFIG.COM | SYS$SYSTEM: STARTUP.COM | Connects special devices and loads device I/O drivers. |
| SYS$MANAGER: SYSECURITY.COM | SYS$SYSTEM: STARTUP.COM | Defines the location of the security audit and archive files before it starts the security audit server. |
| SYS$MANAGER: SYLOGICALS.COM | SYS$SYSTEM: ST ARTUP.COM | Creates systemwide logical names, and defines system components as executive-mode logical names. (Clusterwide logical names should be defined in SYSTARTUP_VMS.COM.) Cluster common disks can be mounted at the end of this procedure. |
| SYS$MANAGER: SYSTARTUP_VMS.COM | SYS$SYSTEM: STARTUP.COM | Performs many of the following startup and login functions:<br>• Mounts all volumes except the system disk.<br>• Sets device characteristics.<br>• Defines clusterwide logical names<br>• Initializes and starts batch and print queues.<br>• Installs known images.<br>• Starts layered products.<br>• Starts the DECnet software.<br>• Analyzes most recent system failure.<br>• Purges old operator log files.<br>• Starts the LAT network (if used).<br>• Defines the maximum number of interactive users.<br>• Announces that the system is up and running.<br>• Allows users to log in. |

The directory SYS$COMMON:[SYSMGR] contains a template file for each command procedure that you can edit. Use the command procedure templates (in SYS$COMMON:[SYSMGR]*.TEMPLATE) as examples for customization of your system's startup and login characteristics.

### 5.6.2 Building Startup Procedures

The first step in preparing an OpenVMS Cluster shared environment is to build a SYSTARTUP_VMS command procedure. Each computer executes the procedure at startup time to define the operating environment.

Prepare the SYSTARTUP_VMS.COM procedure as follows:

| Step | Action |
|---|---|
| 1 | In each computer's SYS$SPECIFIC:[SYSMGR] directory, edit the SYSTARTUP_VMS.TEMPLATE file to set up a SYSTARTUP_VMS.COM procedure that:<br>• Performs computer-specific startup functions such as the following:<br>– Setting up dual-ported and local disks<br>– Loading device drivers<br>– Setting up local terminals and terminal server access<br>• Invoking the common startup procedure (described next). |
| 2 | Build a common command procedure that includes startup commands that you want to be common to all computers. The common procedure might contain commands that:<br>• Install images<br>• Define logical names<br>• Set up queues<br>• Set up and mount physically accessible mass storage devices<br>• Perform any other common startup functions<br>**Note:** You might choose to build these commands into individual command procedures that are invoked from the common procedure. For example, the MSCPMOUNT.COM file in the SYS$EXAMPLES directory is a sample common command procedure that contains commands typically used to mount cluster disks. The example includes comments explaining each phase of the procedure. |
| 3 | Place the common procedure in the SYS$COMMON:[SYSMGR] directory on a common system disk or other cluster-accessible disk.<br>**Important:** The common procedure is usually located in the SYS$COMMON:[SYSMGR] directory on a common system disk but can reside on any disk, provided that the disk is cluster accessible and is mounted when the procedure is invoked. If you create a copy of the common procedure for each computer, you must remember to update each copy whenever you make changes. |

### 5.6.3 Combining Existing Procedures

To build startup procedures for an OpenVMS Cluster system in which existing computers are to be combined, you should compare both the computer-specific SYSTARTUP_VMS and the common startup command procedures on each computer and make any adjustments required. For example, you can compare the procedures from each computer and include commands that define the same logical names in your common SYSTARTUP_VMS command procedure.

After you have chosen which commands to make common, you can build the common procedures on one of the OpenVMS Cluster computers.

### 5.6.4 Using Multiple Startup Procedures

To define a multiple-environment cluster, you set up computer-specific versions of one or more system files. For example, if you want to give users larger working set quotas on URANUS, you would create a computer-specific version of SYSUAF.DAT and place that file in URANUS's SYS$SPECIFIC:[SYSEXE] directory. That directory can be located in URANUS's root on a common system disk or on an individual system disk that you have set up on URANUS.

Follow these steps to build SYSTARTUP and SYLOGIN command files for a multiple-environment OpenVMS Cluster:

| Step | Action |
|---|---|
| 1 | Include in SYSTARTUP_VMS.COM elements that you want to remain unique to a computer, such as commands to define computer-specific logical names and symbols. |
| 2 | Place these files in the SYS$SPECIFIC root on each computer. |

**Example:** Consider a three-member cluster consisting of computers JUPITR, SATURN, and PLUTO. The timesharing environments on JUPITR and SATURN are the same. However, PLUTO runs applications for a specific user group. In this cluster, you would create a common SYSTARTUP_VMS command procedure for JUPITR and SATURN that defines identical environments on these computers. But the command procedure for PLUTO would be different; it would include commands to define PLUTO's special application environment.

## 5.7 Providing OpenVMS Cluster System Security

The OpenVMS security subsystem ensures that all authorization information and object security profiles are consistent across all nodes in the cluster. The OpenVMS VAX and OpenVMS Alpha operating systems do not support multiple security domains because the operating system cannot enforce a level of separation needed to support different security domains on separate cluster members.

### 5.7.1 Security Checks

In an OpenVMS Cluster system, individual nodes use a common set of authorizations to mediate access control that, in effect, ensures that a security check results in the same answer from any node in the cluster. The following list outlines how the OpenVMS operating system provides a basic level of protection:

- Authorized users can have processes executing on any OpenVMS Cluster member.
- A process, acting on behalf of an authorized individual, requests access to a cluster object.
- A coordinating node determines the outcome by comparing its copy of the common authorization database with the security profile for the object being accessed.

The OpenVMS operating system provides the same strategy for the protection of files and queues, and further incorporates all other cluster-visible objects, such as devices, volumes, and lock resource domains.

Starting with OpenVMS Version 7.3, the operating system provides clusterwide intrusion detection, which extends protection against attacks of all types throughout the cluster. The intrusion data and information from each system is integrated to protect the cluster as a whole. Prior to Version 7.3, each system was protected individually.

The SECURITY_POLICY system parameter controls whether a local or a clusterwide intrusion database is maintained for each system. The default setting is for a clusterwide database, which contains all unauthorized attempts and the state of any intrusion events for all cluster members that are using this setting. Cluster members using the clusterwide intrusion database are made aware if a cluster member is under attack or has any intrusion events recorded. Events

recorded on one system can cause another system in the cluster to take restrictive action. (For example, the person attempting to log in is monitored more closely and limited to a certain number of login retries within a limited period of time. Once a person exceeds either the retry or time limitation, he or she cannot log in.)

Actions of the cluster manager in setting up an OpenVMS Cluster system can affect the security operations of the system. You can facilitate OpenVMS Cluster security management using the suggestions discussed in the following sections.

The easiest way to ensure a single security domain is to maintain a single copy of each of the following files on one or more disks that are accessible from anywhere in the OpenVMS Cluster system. When a cluster is configured with multiple system disks, you can use system logical names (as shown in Section 5.10) to ensure that only a single copy of each file exists.

The OpenVMS security domain is controlled by the data in the following files:

SYS$MANAGER:VMS$AUDIT_SERVER.DAT
SYS$SYSTEM:NETOBJECT.DAT
SYS$SYSTEM:NETPROXY.DAT
TCPIP$PROXY.DAT
SYS$SYSTEM:QMAN$MASTER.DAT
SYS$SYSTEM:RIGHTSLIST.DAT
SYS$SYSTEM:SYSALF.DAT
SYS$SYSTEM:SYSUAF.DAT
SYS$SYSTEM:SYSUAFALT.DAT
SYS$SYSTEM:VMS$OBJECTS.DAT
SYS$SYSTEM:VMS$PASSWORD_HISTORY.DATA
SYS$SYSTEM:VMSMAIL_PROFILE.DATA
SYS$LIBRARY:VMS$PASSWORD_DICTIONARY.DATA
SYS$LIBRARY:VMS$PASSWORD_POLICY.EXE

**Note:** Using shared files is not the only way of achieving a single security domain. You may need to use multiple copies of one or more of these files on different nodes in a cluster. For example, on Alpha nodes you may choose to deploy system-specific user authorization files (SYSUAFs) to allow for different memory management working-set quotas among different nodes. Such configurations are fully supported as long as the security information available to each node in the cluster is identical.

## 5.8 Files Relevant to OpenVMS Cluster Security

Table 5–3 describes the security-relevant portions of the files that must be common across all cluster members to ensure that a single security domain exists.

**Notes:**

- Some of these files are created only on request and may not exist in all configurations.
- A file can be absent on one node only if it is absent on all nodes.
- As soon as a required file is created on one node, it must be created or commonly referenced on all remaining cluster nodes.

The following table describes designations for the files in Table 5–3.

| Table Keyword | Meaning |
|---|---|
| Required | The file contains some data that must be kept common across all cluster members to ensure that a single security environment exists. |
| Recommended | The file contains data that should be kept common at the discretion of the site security administrator or system manager. Nonetheless, Digital recommends that you synchronize the recommended files. |

**Table 5–3 Security Files**

| File Name | Contains |
|---|---|
| VMS$AUDIT_SERVER.DAT [recommended] | Information related to security auditing. Among the information contained is the list of enabled security auditing events and the destination of the system security audit journal file. When more than one copy of this file exists, all copies should be updated after any SET AUDIT command.<br><br>OpenVMS Cluster system managers should ensure that the name assigned to the security audit journal file resolves to the following location:<br><br>`SYS$COMMON:[SYSMGR]SECURITY.AUDIT$JOURNAL`<br><br>**Rule:** If you need to relocate the audit journal file somewhere other than the system disk (or if you have multiple system disks), you should redirect the audit journal uniformly across all nodes in the cluster. Use the command SET AUDIT/JOURNAL=SECURITY/DESTINATION=*file-name*, specifying a file name that resolves to the same file throughout the cluster.<br><br>Changes are automatically made in the audit server database, SYS$MANAGER:VMS$AUDIT_SERVER.DAT. This database also identifies which events are enabled and how to monitor the audit system's use of resources, and restores audit system settings each time the system is rebooted.<br><br>**Caution:** Failure to synchronize multiple copies of this file properly may result in partitioned auditing domains.<br><br>**Reference:** For more information, see the *OpenVMS Guide to System Security*. |
| NETOBJECT.DAT [required] | The DECnet object database. Among the information contained in this file is the list of known DECnet server accounts and passwords. When more than one copy of this file exists, all copies must be updated after every use of the NCP commands SET OBJECT or DEFINE OBJECT.<br><br>**Caution:** Failure to synchronize multiple copies of this file properly may result in unexplained network login failures and unauthorized network access. For instructions on maintaining a single copy, refer to Section 5.10.1.<br><br>**Reference:** Refer to the DECnet–Plus documentation for equivalent NCL command information. |
| NETPROXY.DAT and NET$PROXY.DAT [required] | The network proxy database. It is maintained by the OpenVMS Authorize utility. When more than one copy of this file exists, all copies must be updated after any UAF proxy command.<br><br>**Note:** The NET$PROXY.DAT and NETPROXY.DAT files are equivalent; NET$PROXY is for DECnet–Plus implementations and NETPROXY.DAT is for DECnet for OpenVMS implementations.<br><br>**Caution:** Failure to synchronize multiple copies of this file properly may result in unexplained network login failures and unauthorized network access. For instructions on maintaining a single copy, refer to Section 5.10.1.<br><br>**Reference:** Appendix B discusses how to consolidate several NETPROXY.DAT and RIGHTSLIST.DAT files. |

(continued on next page)

**Table 5–3 (Cont.) Security Files**

| File Name | Contains |
|---|---|
| TCPIP$PROXY.DAT | This database provides OpenVMS identities for remote NFS clients and UNIX-style identifiers for local NFS client users; provides proxy accounts for remote processes. For more information about this file, see the *Compaq TCP/IP Services for OpenVMS Management* manual. |
| QMAN$MASTER.DAT [required] | The master queue manager database. This file contains the security information for all shared batch and print queues.<br>**Rule:** If two or more nodes are to participate in a shared queuing system, a single copy of this file must be maintained on a shared disk. For instructions on maintaining a single copy, refer to Section 5.10.1. |
| RIGHTSLIST.DAT [required] | The rights identifier database. It is maintained by the OpenVMS Authorize utility and by various rights identifier system services. When more than one copy of this file exists, all copies must be updated after any change to any identifier or holder records.<br>**Caution:** Failure to synchronize multiple copies of this file properly may result in unauthorized system access and unauthorized access to protected objects. For instructions on maintaining a single copy, refer to Section 5.10.1.<br>**Reference:** Appendix B discusses how to consolidate several NETPROXY.DAT and RIGHTSLIST.DAT files. |
| SYSALF.DAT [required] | The system Autologin facility database. It is maintained by the OpenVMS SYSMAN utility. When more than one copy of this file exists, all copies must be updated after any SYSMAN ALF command.<br>**Note:** This file may not exist in all configurations.<br>**Caution:** Failure to synchronize multiple copies of this file properly may result in unexplained login failures and unauthorized system access. For instructions on maintaining a single copy, refer to Section 5.10.1. |

(continued on next page)

**Table 5–3 (Cont.) Security Files**

| File Name | Contains |
|---|---|
| SYSUAF.DAT [required] | The system user authorization file. It is maintained by the OpenVMS Authorize utility and is modifiable via the $SETUAI system service. When more than one copy of this file exists, you must ensure that the SYSUAF and associated $SETUAI item codes are synchronized for each user record. The following table shows the fields in SYSUAF and their associated $SETUAI item codes. |

| Internal Field Name | $SETUAI Item Code |
|---|---|
| UAF$R_DEF_CLASS | UAI$_DEF_CLASS |
| UAF$Q_DEF_PRIV | UAI$_DEF_PRIV |
| UAF$B_DIALUP_ACCESS_P | UAI$_DIALUP_ACCESS_P |
| UAF$B_DIALUP_ACCESS_S | UAI$_DIALUP_ACCESS_S |
| UAF$B_ENCRYPT | UAI$_ENCRYPT |
| UAF$B_ENCRYPT2 | UAI$_ENCRYPT2 |
| UAF$Q_EXPIRATION | UAI$_EXPIRATION |
| UAF$L_FLAGS | UAI$_FLAGS |
| UAF$B_LOCAL_ACCESS_P | UAI$_LOCAL_ACCESS_P |
| UAF$B_LOCAL_ACCESS_S | UAI$_LOCAL_ACCESS_S |
| UAF$B_NETWORK_ACCESS_P | UAI$_NETWORK_ACCESS_P |
| UAF$B_NETWORK_ACCESS_S | UAI$_NETWORK_ACCESS_S |
| UAF$B_PRIME_DAYS | UAI$_PRIMEDAYS |
| UAF$Q_PRIV | UAI$_PRIV |
| UAF$Q_PWD | UAI$_PWD |
| UAF$Q_PWD2 | UAI$_PWD2 |
| UAF$Q_PWD_DATE | UAI$_PWD_DATE |
| UAF$Q_PWD2_DATE | UAI$_PWD2_DATE |
| UAF$B_PWD_LENGTH | UAI$_PWD_LENGTH |
| UAF$Q_PWD_LIFETIME | UAI$_PWD_LIFETIME |
| UAF$B_REMOTE_ACCESS_P | UAI$_REMOTE_ACCESS_P |
| UAF$B_REMOTE_ACCESS_S | UAI$_REMOTE_ACCESS_S |
| UAF$R_MAX_CLASS | UAI$_MAX_CLASS |
| UAF$R_MIN_CLASS | UAI$_MIN_CLASS |
| UAF$W_SALT | UAI$_SALT |
| UAF$L_UIC | Not applicable |

**Caution:** Failure to synchronize multiple copies of the SYSUAF files properly may result in unexplained login failures and unauthorized system access. For instructions on maintaining a single copy, refer to Section 5.10.1.

**Reference:** Appendix B discusses creation and management of the various elements of an OpenVMS Cluster common SYSUAF.DAT authorization database.

(continued on next page)

**Table 5–3 (Cont.) Security Files**

| File Name | Contains |
|---|---|
| SYSUAFALT.DAT [required] | The system alternate user authorization file. This file serves as a backup to SYSUAF.DAT and is enabled via the SYSUAFALT system parameter. When more than one copy of this file exists, all copies must be updated after any change to any authorization records in this file.<br>**Note:** This file may not exist in all configurations.<br>**Caution:** Failure to synchronize multiple copies of this file properly may result in unexplained login failures and unauthorized system access. |
| †VMS$OBJECTS.DAT [required] | On VAX systems, this file is located in SYS$COMMON:[SYSEXE] and contains the clusterwide object database. Among the information contained in this file are the security profiles for all clusterwide objects. When more than one copy of this file exists, all copies must be updated after any change to the security profile of a clusterwide object or after new clusterwide objects are created. Clusterwide objects include disks, tapes, and resource domains.<br>OpenVMS Cluster system managers should ensure that the security object database is present on each node in the OpenVMS Cluster by specifying a file name that resolves to the same file throughout the cluster, not to a file that is unique to each node.<br>The database is updated whenever characteristics are modified, and the information is distributed so that all nodes participating in the cluster share a common view of the objects. The security database is created and maintained by the audit server process.<br>**Rule:** If you relocate the database, be sure the logical name VMS$OBJECTS resolves to the same file for all nodes in a common-environment cluster. To reestablish the logical name after each system boot, define the logical in SYSECURITY.COM.<br>**Caution:** Failure to synchronize multiple copies of this file properly may result in unauthorized access to protected objects. |
| VMS$PASSWORD_ HISTORY.DATA [recommended] | The system password history database. It is maintained by the system password change facility. When more than one copy of this file exists, all copies should be updated after any password change.<br>**Caution:** Failure to synchronize multiple copies of this file properly may result in a violation of the system password policy. |
| VMSMAIL_PROFILE.DATA [recommended] | The system mail database. This file is maintained by the OpenVMS Mail utility and contains mail profiles for all system users. Among the information contained in this file is the list of all mail forwarding addresses in use on the system. When more than one copy of this file exists, all copies should be updated after any changes to mail forwarding.<br>**Caution:** Failure to synchronize multiple copies of this file properly may result in unauthorized disclosure of information. |
| VMS$PASSWORD_ DICTIONARY.DATA [recommended] | The system password dictionary. The system password dictionary is a list of English language words and phrases that are not legal for use as account passwords. When more than one copy of this file exists, all copies should be updated after any site-specific additions.<br>**Caution:** Failure to synchronize multiple copies of this file properly may result in a violation of the system password policy. |

†VAX specific

(continued on next page)

**Table 5–3 (Cont.) Security Files**

| File Name | Contains |
|---|---|
| VMS$PASSWORD_POLICY.EXE [recommended] | Any site-specific password filters. It is created and installed by the site-security administrator or system manager. When more than one copy of this file exists, all copies should be identical.<br>**Caution:** Failure to synchronize multiple copies of this file properly may result in a violation of the system password policy.<br>**Note:** System managers can create this file as an image to enforce their local password policy. This is an architecture-specific image file that cannot be shared between VAX and Alpha computers. |

## 5.9 Network Security

Network security should promote interoperability and uniform security approaches throughout networks. The following list shows three major areas of network security:

- User authentication
- OpenVMS Cluster membership management
- Using a security audit log file

OpenVMS Cluster system managers should also ensure consistency in the use of DECnet software for intracluster communication.

### 5.9.1 Mechanisms

Depending on the level of network security required, you might also want to consider how other security mechanisms, such as protocol encryption and decryption, can promote additional security protection across the cluster.

**Reference:** See the *OpenVMS Guide to System Security*.

## 5.10 Coordinating System Files

Follow these guidelines to coordinate system files:

| IF you are setting up... | THEN follow the procedures in... |
|---|---|
| A common-environment OpenVMS Cluster that consists of newly installed systems | *OpenVMS System Manager's Manual* to build these files. Because the files on new operating systems are empty except for the Digital-supplied accounts, very little coordination is necessary. |
| An OpenVMS Cluster that will combine one or more computers that have been running with computer-specific files | Appendix B to create common copies of the files from the computer-specific files. |

### 5.10.1 Procedure

In a common-environment cluster with one common system disk, you use a common copy of each system file and place the files in the SYS$COMMON:[SYSEXE] directory on the common system disk or on a disk that is mounted by all cluster nodes. No further action is required.

To prepare a common user environment for an OpenVMS Cluster system that includes more than one common VAX system disk or more than one common Alpha system disk, you must coordinate the system files on those disks.

**Rules:** The following rules apply to the procedures described in Table 5–4:

- Disks holding common resources must be mounted early in the system startup procedure, such as in the SYLOGICALS.COM procedure.
- You must ensure that the disks are mounted with each OpenVMS Cluster reboot.

**Table 5–4 Procedure for Coordinating Files**

| Step | Action |
|---|---|
| **1** | Decide where to locate the SYSUAF.DAT and NETPROXY.DAT files. In a cluster with multiple system disks, system management is much easier if the common system files are located on a single disk that is not a system disk. |
| **2** | Copy SYS$SYSTEM:SYSUAF.DAT and SYS$SYSTEM:NETPROXY.DAT to a location other than the system disk. |
| **3** | Copy SYS$SYSTEM:RIGHTSLIST.DAT and SYS$SYSTEM:VMSMAIL_PROFILE.DATA to the same directory in which SYSUAF.DAT and NETPROXY.DAT reside. |
| **4** | Edit the file SYS$COMMON:[SYSMGR]SYLOGICALS.COM *on each system disk* and define logical names that specify the location of the cluster common files.<br>**Example:** If the files will be located on $1$DJA16, define logical names as follows:<br>`$ DEFINE/SYSTEM/EXEC SYSUAF -`<br>`  $1$DJA16:[VMS$COMMON.SYSEXE]SYSUAF.DAT`<br>`$ DEFINE/SYSTEM/EXEC NETPROXY -`<br>`  $1$DJA16:[VMS$COMMON.SYSEXE]NETPROXY.DAT`<br>`$ DEFINE/SYSTEM/EXEC RIGHTSLIST -`<br>`  $1$DJA16:[VMS$COMMON.SYSEXE]RIGHTSLIST.DAT`<br>`$ DEFINE/SYSTEM/EXEC VMSMAIL_PROFILE -`<br>`  $1$DJA16:[VMS$COMMON.SYSEXE]VMSMAIL_PROFILE.DATA`<br>`$ DEFINE/SYSTEM/EXEC NETNODE_REMOTE -`<br>`  $1$DJA16:[VMS$COMMON.SYSEXE]NETNODE_REMOTE.DAT`<br>`$ DEFINE/SYSTEM/EXEC NETNODE_UPDATE -`<br>`  $1$DJA16:[VMS$COMMON.SYSMGR]NETNODE_UPDATE.COM`<br>`$ DEFINE/SYSTEM/EXEC QMAN$MASTER -`<br>`  $1$DJA16:[VMS$COMMON.SYSEXE]` |
| **5** | To ensure that the system disks are mounted correctly with each reboot, follow these steps:<br>1. Copy the SYS$EXAMPLES:CLU_MOUNT_DISK.COM file to the [VMS$COMMON.SYSMGR] directory, and edit it for your configuration.<br>2. Edit SYLOGICALS.COM and include commands to mount, with the appropriate volume label, the system disk containing the shared files.<br>**Example:** If the system disk is $1$DJA16, include the following command:<br>`$ @SYS$SYSDEVICE:[VMS$COMMON.SYSMGR]CLU_MOUNT_DISK.COM $1$DJA16: volume-label` |
| **6** | When you are ready to start the queuing system, be sure you have moved the queue and journal files to a cluster-available disk. Any cluster common disk is a good choice if the disk has sufficient space.<br>Enter the following command:<br>`$ START/QUEUE/MANAGER $1$DJA16:[VMS$COMMON.SYSEXE]` |

## 5.10.2 Network Database Files

In OpenVMS Cluster systems on the LAN and in mixed-interconnect clusters, you must also coordinate the SYS$MANAGER:NETNODE_UPDATE.COM file, which is a file that contains all essential network configuration data for satellites. NETNODE_UPDATE.COM is updated each time you add or remove a satellite or change its Ethernet or FDDI hardware address. This file is discussed more thoroughly in Section 10.4.2.

In OpenVMS Cluster systems configured with DECnet for OpenVMS software, you must also coordinate NETNODE_REMOTE.DAT, which is the remote node network database.

## 5.11 System Time on the Cluster

When a computer joins the cluster, the cluster attempts to set the joining computer's system time to the current time on the cluster. Although it is likely that the system time will be similar on each cluster computer, there is no assurance that the time will be set. Also, no attempt is made to ensure that the system times remain similar throughout the cluster. (For example, there is no protection against different computers having different clock rates.)

An OpenVMS Cluster system spanning multiple time zones must use a single, clusterwide common time on all nodes. Use of a common time ensures timestamp consistency (for example, between applications, file-system instances) across the OpenVMS Cluster members.

### 5.11.1 Setting System Time

Use the SYSMAN command CONFIGURATION SET TIME to set the time across the cluster. This command issues warnings if the time on all nodes cannot be set within certain limits. Refer to the *OpenVMS System Manager's Manual* for information about the SET TIME command.

# 6
# Cluster Storage Devices

One of the most important features of OpenVMS Cluster systems is the ability to provide access to devices and files across multiple systems.

In a traditional computing environment, a single system is directly attached to its storage subsystems. Even though the system may be networked with other systems, when the system is shut down, no other system on the network has access to its disks or any other devices attached to the system.

In an OpenVMS Cluster system, disks and tapes can be made accessible to one or more members. So, if one computer shuts down, the remaining computers still have access to the devices.

## 6.1 Data File Sharing

Cluster-accessible devices play a key role in OpenVMS Clusters because, when you place data files or applications on a cluster-accessible device, computers can share a single copy of each common file. Data sharing is possible between VAX computers, between Alpha computers, and between VAX and Alpha computers.

In addition, multiple systems (VAX and Alpha) can write to a shared disk file simultaneously. It is this ability that allows multiple systems in an OpenVMS Cluster to share a single system disk; multiple systems can boot from the same system disk and share operating system files and utilities to save disk space and simplify system management.

**Note:** Tapes do not allow multiple systems to access a tape file simultaneously.

### 6.1.1 Access Methods

Depending on your business needs, you may want to restrict access to a particular device to the users on the computer that are directly connected (local) to the device. Alternatively, you may decide to set up a disk or tape as a served device so that any user on any OpenVMS Cluster computer can allocate and use it.

Table 6–1 describes the various access methods.

**Table 6–1 Device Access Methods**

| Method | Device Access | Comments | Illustrated in |
|---|---|---|---|
| Local | Restricted to the computer that is directly connected to the device. | Can be set up to be served to other systems. | Figure 6–3 |
| Dual ported | Using either of two physical ports, each of which can be connected to separate controllers. A dual-ported disk can survive the failure of a single controller by failing over to the other controller. | As long as one of the controllers is available, the device is accessible by all systems in the cluster. | Figure 6–1 |
| Shared | Through a shared interconnect to multiple systems. | Can be set up to be served to systems that are not on the shared interconnect. | Figure 6–2 |
| Served | Through a computer that has the MSCP or TMSCP server software loaded. | MSCP and TMSCP serving are discussed in Section 6.3. | Figures 6–2 and 6–3 |
| Dual pathed | Possible through more than one path. | If one path fails, the device is accessed over the other path. Requires the use of allocation classes (described in Section 6.2.1 to provide a unique, path-independent name.) | Figure 6–2 |

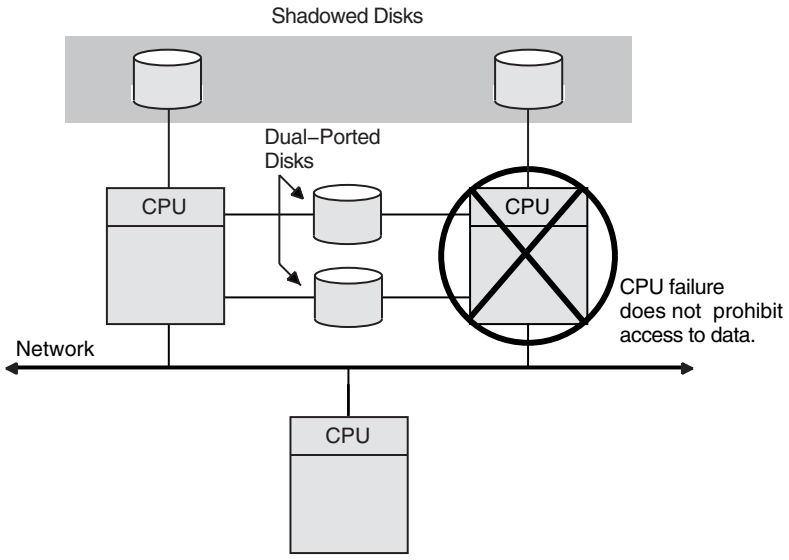
**Note:** The path to an individual disk may appear to be local from some nodes and served from others.

### 6.1.2 Examples

When storage subsystems are connected directly to a specific system, the availability of the subsystem is lower due to the reliance on the host system. To increase the availability of these configurations, OpenVMS Cluster systems support dual porting, dual pathing, and MSCP and TMSCP serving.

Figure 6–1 shows a dual-ported configuration, in which the disks have independent connections to two separate computers. As long as one of the computers is available, the disk is accessible by the other systems in the cluster.

**Figure 6–1 Dual-Ported Disks**

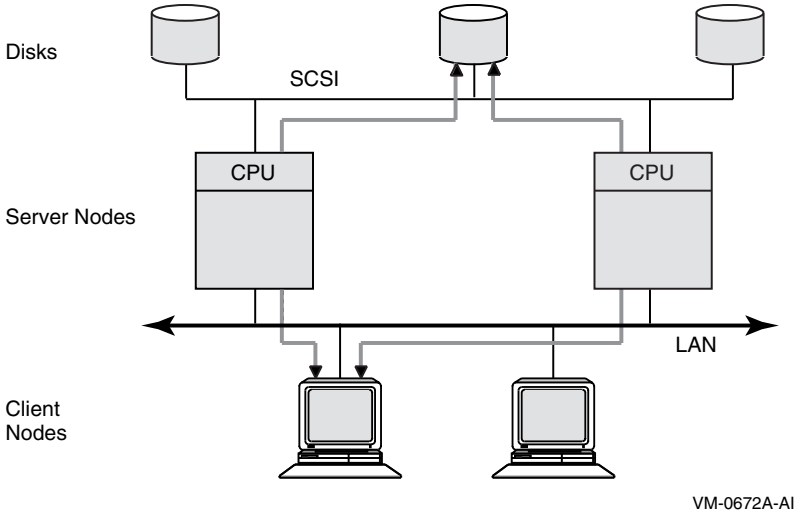



**Note:** Disks can also be shadowed using Volume Shadowing for OpenVMS. The automatic recovery from system failure provided by dual porting and shadowing is transparent to users and does not require any operator intervention.

Figure 6–2 shows a dual-pathed DSSI and Ethernet configuration. The disk devices, accessible through a shared SCSI interconnect, are MSCP served to the client nodes on the LAN.

**Rule:** A dual-pathed DSA disk cannot be used as a system disk for a directly connected CPU. Because a device can be on line to one controller at a time, only one of the server nodes can use its local connection to the device. The second server node accesses the device through the MSCP (or the TMSCP server). If the computer that is currently serving the device fails, the other computer detects the failure and fails the device over to its local connection. The device thereby remains available to the cluster.

**Figure 6–2 Dual-Pathed Disks**



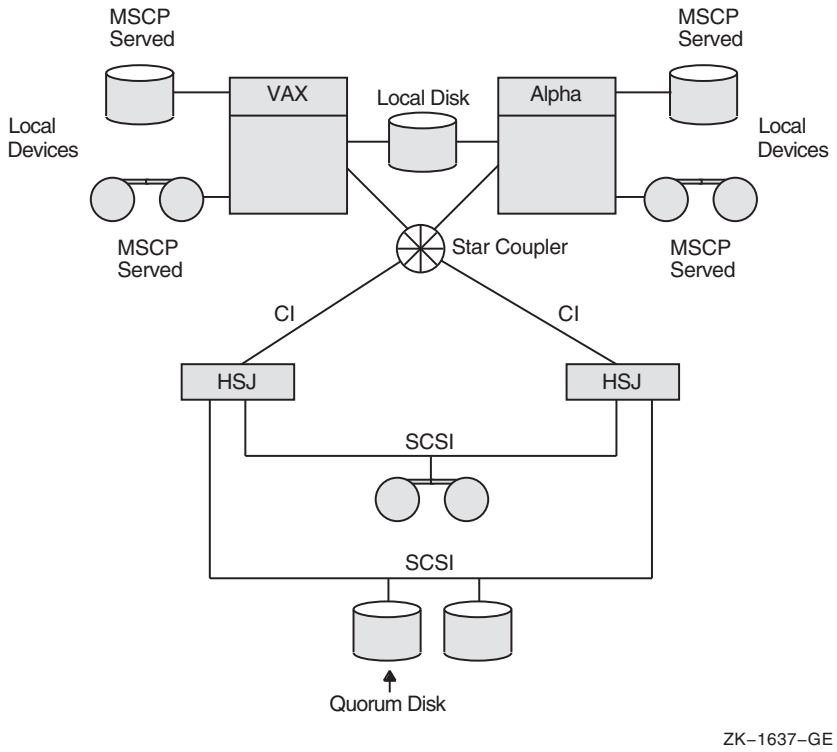

Dual-pathed disks or tapes can be failed over between two computers that serve the devices to the cluster, provided that:

- The same device controller letter is generated and the same allocation class is specified on each computer, with the result that the device has the same name on both systems. (Section 6.2.1 describes allocation classes.)
- Both computers are running the MSCP server for disks, the TMSCP server for tapes, or both.

**Caution:** Failure to observe these requirements can endanger data integrity.

You can set up HSC or HSJ storage devices to be dual ported between two storage subsystems, as shown in Figure 6–3.

**Figure 6–3 Configuration with Cluster-Accessible Devices**




By design, HSC and HSJ disks and tapes are directly accessible by all OpenVMS Cluster nodes that are connected to the same star coupler. Therefore, if the devices are dual ported, they are automatically dual pathed. Computers connected by CI can access a dual-ported HSC or HSJ device by way of a path through either subsystem connected to the device. If one subsystem fails, access fails over to the other subsystem.

**Note:** To control the path that is taken during failover, you can specify a preferred path to force access to disks over a specific path. Section 6.1.3 describes the preferred-path capability.

### 6.1.3 Specifying a Preferred Path

The operating system supports specifying a preferred path for DSA disks, including RA series disks and disks that are accessed through the MSCP server. (This function is not available for tapes.) If a preferred path is specified for a disk, the MSCP disk class drivers use that path:

- For the first attempt to locate the disk and bring it on line with a DCL command MOUNT
- For failover of an already mounted disk

In addition, you can initiate failover of a mounted disk to force the disk to the preferred path or to use load-balancing information for disks accessed by MSCP servers.

You can specify the preferred path by using the SET PREFERRED_PATH DCL command or by using the $QIO function (IO$_SETPRFPATH), with the P1 parameter containing the address of a counted ASCII string (.ASCIC). This string

is the node name of the HSC or HSJ, or of the OpenVMS system that is to be the preferred path.

**Rule:** The node name must match an existing node running the MSCP server that is known to the local node.

**Reference:** For more information about the use of the SET PREFERRED_PATH DCL command, refer to the *OpenVMS DCL Dictionary: N–Z*.

For more information about the use of the IO$_SETPRFPATH function, refer to the *OpenVMS I/O User's Reference Manual*.

## 6.2 Naming OpenVMS Cluster Storage Devices

In the OpenVMS operating system, a device name takes the form of *ddcu*, where:

- *dd* represents the predefined code for the device type
- *c* represents the predefined controller designation
- *u* represents the unit number

For CI or DSSI devices, the controller designation is always the letter A; and the unit number is selected by the system manager.

For SCSI devices, the controller letter is assigned by OpenVMS, based on the system configuration. The unit number is determined by the SCSI bus ID and the logical unit number (LUN) of the device.

Because device names must be unique in an OpenVMS Cluster, and because every cluster member must use the same name for the same device, OpenVMS adds a prefix to the device name, as follows:

- If a device is attached to a single computer, the device name is extended to include the name of that computer:

  ```
  node$ddcu
  ```

  where *node* represents the SCS node name of the system on which the device resides.

- If a device is attached to multiple computers, the node name part of the device name is replaced by a dollar sign and a number (called a node or port allocation class, depending on usage), as follows:

  ```
  $allocation-class$ddcu
  ```

### 6.2.1 Allocation Classes

The purpose of allocation classes is to provide unique and unchanging device names. The device name is used by the OpenVMS Cluster distributed lock manager in conjunction with OpenVMS facilities (such as RMS and the XQP) to uniquely identify shared devices, files, and data.

Allocation classes are required in OpenVMS Cluster configurations where storage devices are accessible through multiple paths. Without the use of allocation classes, device names that relied on node names would change as access paths to the devices change.

Prior to OpenVMS Version 7.1, only one type of allocation class existed, which was node based. It was named **allocation class**. OpenVMS Version 7.1 introduced a second type, **port allocation class**, which is specific to a single interconnect and is assigned to all devices attached to that interconnect. Port allocation classes were originally designed for naming SCSI devices. Their use has been expanded

to include additional devices types: floppy disks, PCI RAID controller disks, and IDE disks.

The use of port allocation classes is optional. They are designed to solve the device-naming and configuration conflicts that can occur in certain configurations, as described in Section 6.2.3.

To differentiate between the earlier node-based allocation class and the newer port allocation class, the term **node allocation class** was assigned to the earlier type.

Prior to OpenVMS Version 7.2, all nodes with direct access to the same multipathed device were required to use the same nonzero value for the node allocation class. OpenVMS Version 7.2 introduced the MSCP_SERVE_ALL system parameter, which can be set to serve all disks or to exclude those whose node allocation class differs.

**Note**

If SCSI devices are connected to multiple hosts and if port allocation classes are *not* used, then all nodes with direct access to the same multipathed devices must use the same nonzero node allocation class.

Multipathed MSCP controllers also have an allocation class parameter, which is set to match that of the connected nodes. (If the allocation class does not match, the devices attached to the nodes cannot be served.)

### 6.2.2 Specifying Node Allocation Classes

A node allocation class can be assigned to computers, HSC or HSJ controllers, and DSSI ISEs. The node allocation class is a numeric value from 1 to 255 that is assigned by the system manager.

The default node allocation class value is 0. A node allocation class value of 0 is appropriate only when serving a local, single-pathed disk. If a node allocation class of 0 is assigned, served devices are named using the *node-name*$*device-name* syntax, that is, the device name prefix reverts to the node name.

The following rules apply to specifying node allocation class values:

1. When serving satellites, the same nonzero node allocation class value must be assigned to the serving computers and controllers.
2. All cluster-accessible devices on computers with a nonzero node allocation class value must have unique names throughout the cluster. For example, if two computers have the same node allocation class value, it is invalid for both computers to have a local disk named DJA0 or a tape named MUA0. This also applies to HSC and HSJ subsystems.

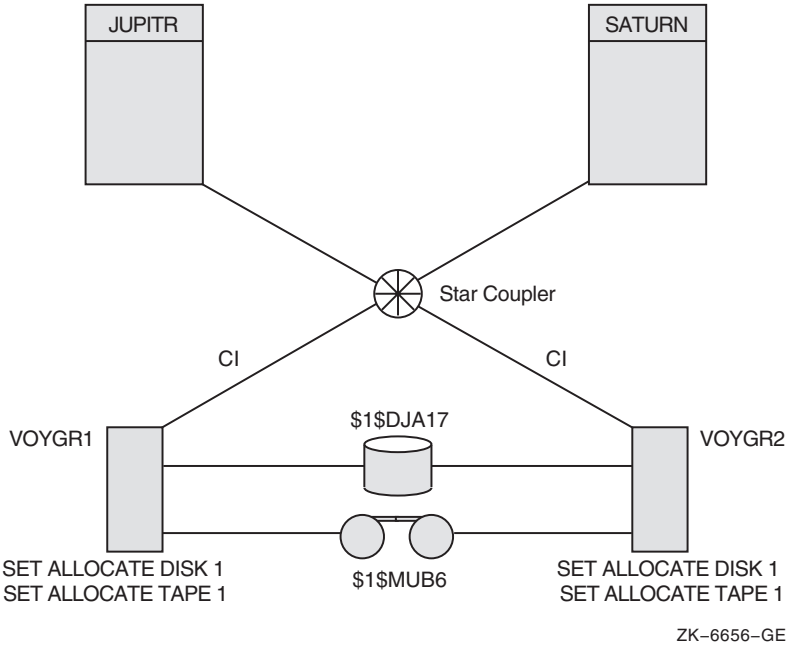
System managers provide node allocation classes separately for disks and tapes. The node allocation class for disks and the node allocation class for tapes can be different.

The node allocation class names are constructed as follows:

```
$disk-allocation-class$device-name
$tape-allocation-class$device-name
```

**Caution:** Failure to set node allocation class values and device unit numbers correctly can endanger data integrity and cause locking conflicts that suspend normal cluster operations. Figure 6–4 shows an example of how cluster device names are specified in a CI configuration.

**Figure 6–4 Disk and Tape Dual Pathed Between HSC Controllers**




In this configuration:

- The disk device name ($1$DJA17) and tape device name ($1$MUB6) are derived using the node allocation class of the two controllers.
- Node allocation classes are not required for the two computers.
- JUPITR and SATURN can access the disk or tape through either VOYGR1 or VOYGR2.
- Note that the disk and tape allocation classes do not need to be the same.

If one controller with node allocation class 1 is not available, users can gain access to a device specified by that node allocation class through the other controller.

Figure 6–6 builds on Figure 6–4 by including satellite nodes that access devices $1$DUA17 and $1$MUA12 through the JUPITR and NEPTUN computers. In this configuration, the computers JUPITR and NEPTUN require node allocation classes so that the satellite nodes are able to use consistent device names regardless of the access path to the devices.

**Note:** System management is usually simplified by using the same node allocation class value for all servers, HSC and HSJ subsystems, and DSSI ISEs; you can arbitrarily choose a number between 1 and 255. Note, however, that to change a node allocation class value, you must shut down and reboot the entire cluster (described in Section 8.6). If you use a common node allocation class for computers and controllers, ensure that all devices have unique unit numbers.

#### 6.2.2.1 Assigning Node Allocation Class Values on Computers

There are two ways to assign a node allocation class: by using CLUSTER_CONFIG.COM or CLUSTER_CONFIG_LAN.COM, which is described in Section 8.4, or by using AUTOGEN, as shown in the following table.

| Step | Action |
|---|---|
| **1** | Edit the root directory [SYS*n*.SYSEXE]MODPARAMS.DAT on each node that boots from the system disk. The following example shows a MODPARAMS.DAT file. The entries are hypothetical and should be regarded as examples, not as suggestions for specific parameter settings.<br><br>`!`<br>`! Site-specific AUTOGEN data file.  In an OpenVMS Cluster`<br>`! where a common system disk is being used, this file`<br>`! should reside in SYS$SPECIFIC:[SYSEXE], not a common`<br>`! system directory.`<br>`!`<br>`! Add modifications that you want to make to AUTOGEN's`<br>`! hardware configuration data, system parameter`<br>`! calculations, and page, swap, and dump file sizes`<br>`! to the bottom of this file.`<br><br>`SCSNODE="NODE01"`<br>`SCSSYSTEMID=99999`<br>`NISCS_LOAD_PEA0=1`<br>`VAXCLUSTER=2`<br>`MSCP_LOAD=1`<br>`MSCP_SERVE_ALL=1`<br>`ALLOCLASS=1`<br>`TAPE_ALLOCLASS=1` |
| **2** | Invoke AUTOGEN to set the system parameter values:<br><br>`$ @SYS$UPDATE:AUTOGEN start-phase end-phase` |
| **3** | Shut down and reboot the entire cluster in order for the new values to take effect. |

#### 6.2.2.2 Assigning Node Allocation Class Values on HSC Subsystems

Assign or change node allocation class values on HSC subsystems while the cluster is shut down. To assign a node allocation class to disks for an HSC subsystem, specify the value using the HSC console command in the following format:

SET ALLOCATE DISK allocation-class-value

To assign a node allocation class for tapes, enter a SET ALLOCATE TAPE command in the following format:

SET ALLOCATE TAPE tape-allocation-class-value

For example, to change the value of a node allocation class for disks to 1, set the HSC internal door switch to the Enable position and enter a command sequence like the following at the appropriate HSC consoles:

```
Ctrl/C
HSC> RUN SETSHO
SETSHO> SET ALLOCATE DISK 1
SETSHO> EXIT
SETSHO-Q Rebooting HSC; Y to continue, Ctrl/Y to abort:? Y
```

Restore the HSC internal door-switch setting.

**Reference:** For complete information about the HSC console commands, refer to the HSC hardware documentation.

#### 6.2.2.3 Assigning Node Allocation Class Values on HSJ Subsystems

To assign a node allocation class value for disks for an HSJ subsystem, enter a SET CONTROLLER MSCP_ALLOCATION_CLASS command in the following format:

SET CONTROLLER MSCP_ALLOCATION_CLASS = allocation-class-value

To assign a node allocation class value for tapes, enter a SET CONTROLLER TMSCP_ALLOCATION_CLASS ALLOCATE TAPE command in the following format:

SET CONTROLLER TMSCP_ALLOCATION_CLASS = allocation-class-value

For example, to assign or change the node allocation class value for disks to 254 on an HSJ subsystem, use the following command at the HSJ console prompt (PTMAN>):

```
PTMAN> SET CONTROLLER MSCP_ALLOCATION_CLASS = 254
```

#### 6.2.2.4 Assigning Node Allocation Class Values on HSD Subsystems

To assign or change allocation class values on any HSD subsystem, use the following commands:

```
$ MC SYSMAN IO CONNECT FYA0:/NOADAP/DRIVER=SYS$FYDRIVER
$ SET HOST/DUP/SERVER=MSCP$DUP/TASK=DIRECT node-name
$ SET HOST/DUP/SERVER=MSCP$DUP/TASK=PARAMS node-name
PARAMS> SET FORCEUNI 0
PARAMS> SET ALLCLASS 143
PARAMS> SET UNITNUM 900
PARAMS> WRITE
Changes require controller initialization, ok? [Y/(N)] Y
PARAMS> EXIT
$
```

#### 6.2.2.5 Assigning Node Allocation Class Values on DSSI ISEs

To assign or change node allocation class values on any DSSI ISE, the command you use differs depending on the operating system.

For example, to change the allocation class value to 1 for a DSSI ISE TRACER, follow the steps in Table 6–2.

**Table 6–2 Changing a DSSI Allocation Class Value**

| Step | Task |
|---|---|
| **1** | Log into the SYSTEM account on the computer connected to the hardware device TRACER and load its driver as follows:<br>• If the computer is an Alpha system, then enter the following command at the DCL prompt:<br>`$ MC SYSMAN IO CONN FYA0:/NOADAP/DRIVER=SYS$FYDRIVER`<br>• If the computer is a VAX system, then enter the following command at the DCL prompt:<br>`$ MCR SYSGEN CONN FYA0:/NOADAP/DRIVER=FYDRIVER` |

(continued on next page)

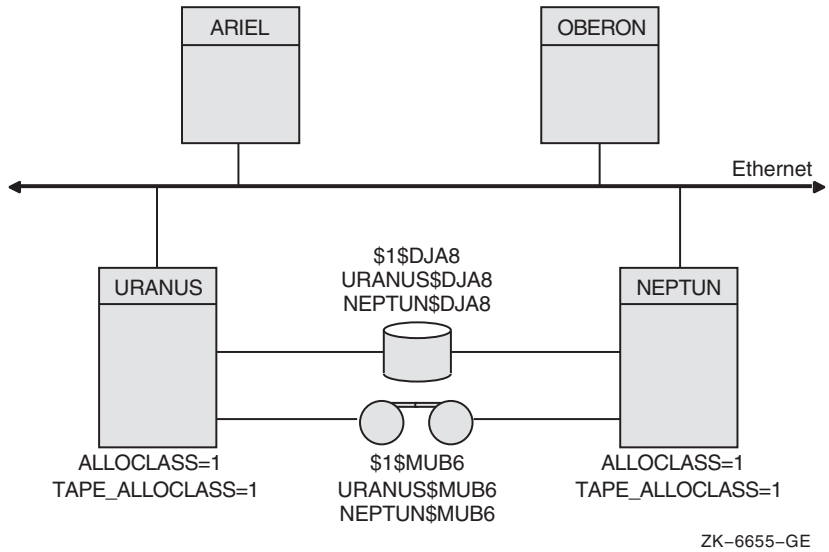
**Table 6–2 (Cont.) Changing a DSSI Allocation Class Value**

| Step | Task |
|---|---|
| **2** | At the DCL prompt, enter the SHOW DEVICE FY command to verify the presence and status of the FY device, as follows:<br>`$ SHOW DEVICE FY`<br>`Device  Device  Error`<br>`Name    Status  Count`<br>`FYA0:   offline     0` |
| **3** | At the DCL prompt, enter the following command sequence to set the allocation class value to 1:<br>`$ SET HOST/DUP/SERVER=MSCP$DUP/TASK=PARAMS` *node-name*<br>`params >set allclass 1`<br>`params >write`<br>`Changes require controller initialization, ok?[Y/N]Y`<br>`Initializing...`<br>`%HSCPAD-S-REMPGMEND, Remote program terminated--message number 3.`<br>`%PAxx, Port has closed virtual circuit - remote node TRACER`<br>`%HSCPAD-S-END, control returned to node` *node-name*<br>`$` |
| **4** | Reboot the entire cluster in order for the new value to take effect. |

#### 6.2.2.6 Node Allocation Class Example With a DSA Disk and Tape

Figure 6–5 shows a DSA disk and tape that are dual pathed between two computers.

**Figure 6–5 Disk and Tape Dual Pathed Between Computers**




In this configuration:

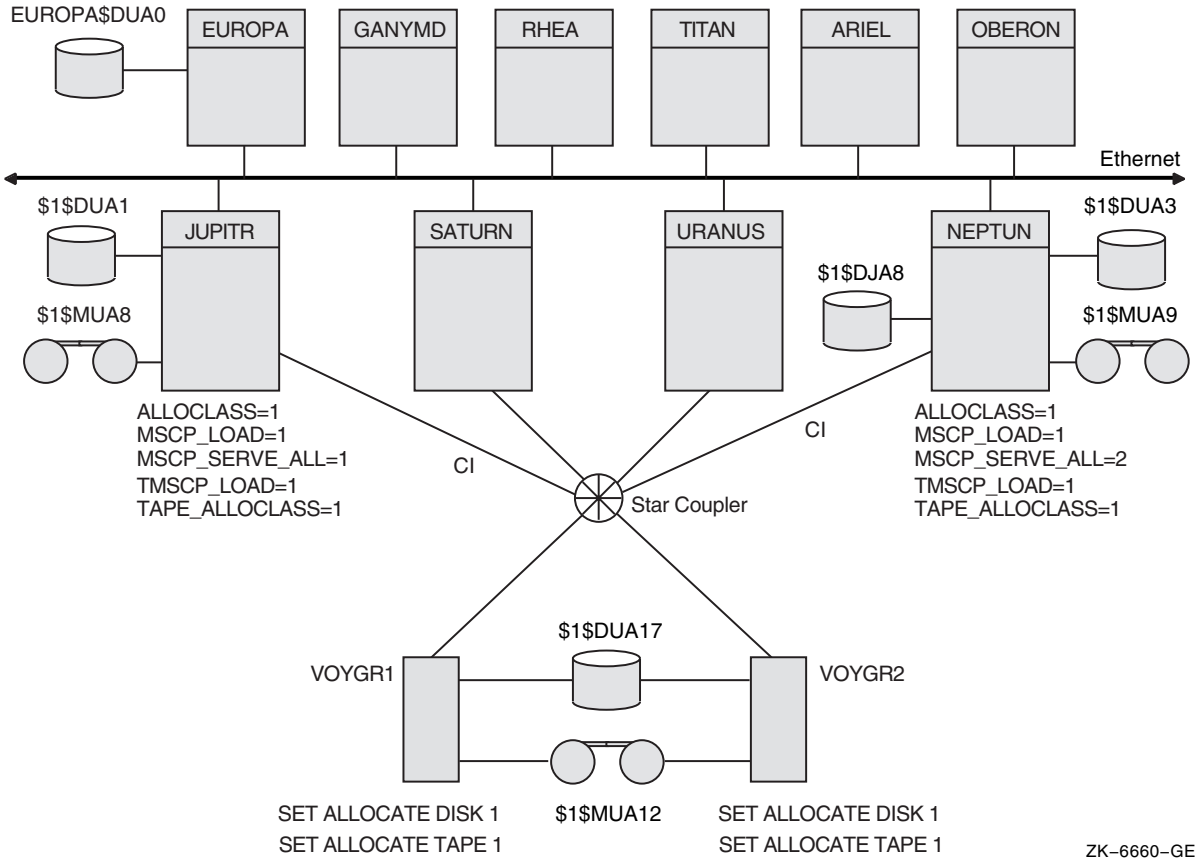
- URANUS and NEPTUN access the disk either locally or through the other computer's MSCP server.

- When satellites ARIEL and OBERON access $1$DJA8, a path is made through either URANUS or NEPTUN.
- If, for example, the node URANUS has been shut down, the satellites can access the devices through NEPTUN. When URANUS reboots, access is available through either URANUS or NEPTUN.

#### 6.2.2.7 Node Allocation Class Example With Mixed Interconnects

Figure 6–6 shows how device names are typically specified in a mixed-interconnect cluster. This figure also shows how relevant system parameter values are set for each CI computer.

**Figure 6–6 Device Names in a Mixed-Interconnect Cluster**




In this configuration:

- A disk and a tape are dual pathed to the HSC or HSJ subsystems named VOYGR1 and VOYGR2; these subsystems are connected to JUPITR, SATURN, URANUS and NEPTUN through the star coupler.
- The MSCP and TMSCP servers are loaded on JUPITR and NEPTUN (MSCP_LOAD = 1, TMSCP_LOAD = 1) and the ALLOCLASS and TAPE_ALLOCLASS parameters are set to the same value (1) on these computers and on both HSC or HSJ subsystems.

**Note:** For optimal availability, two or more CI connected computers should serve HSC or HSJ devices to the cluster.

#### 6.2.2.8 Node Allocation Classes and VAX 6000 Tapes

You must ensure that any tape drive is identified by a unique name that includes a tape allocation class so that naming conflicts do not occur.

##### Avoiding Duplicate Names

Duplicate names are more probable with VAX 6000 computers because TK console tape drives (located in the VAX 6000 cabinet) are usually named either MUA6 or MUB6. Thus, when you configure a VAXcluster system with more than one VAX 6000 computer, multiple TK console tape drives are likely to have the same name.

##### Specifying a Tape Allocation Class

To ensure that the TK console tape drives have names that are unique across the cluster, specify a tape allocation class name as a numeric value from 0 to 255, followed by the device name, as follows:

*$tape-allocation-class$device-name*

**Example:**

Assume that $1$MUA6, $1$MUB6, $2$MUA6 are all unique device names. The first two have the same tape allocation class but have different controller letters (A and B, respectively). The third device has a different tape allocation class than the first two.

##### Ensuring a Unique Access Path

Consider the methods described in Table 6–3 to ensure a unique access path to VAX 6000 TK console tape drives.

**Table 6–3 Ensuring Unique Tape Access Paths**

| Method | Description | Comments |
|---|---|---|
| Set the TK console tape unit number to a unique value on each VAX 6000 system. | For VAXcluster systems in which tapes must be TMSCP served across the cluster, the tape controller letter and unit number of these tape drives must be unique clusterwide and must conform to the cluster device-naming conventions. If controller letters and unit numbers are unique clusterwide, the TAPE_ ALLOCLASS system parameter can be set to the same value on multiple VAX 6000 systems. | The unit number of the TK console drives is controlled by the BI bus unit number plug of the TBK70 controller in the VAX 6000 BI backplane. A Compaq services technician should change the unit number so that it is unique from all other controller cards in the BI backplane. The unit numbers available are in the range of 0 to 15 (the default value is 6). |
| For VAXcluster systems configured with two or more VAX 6000 computers, set up the console tapes with different controller letters. | If your VAXcluster configuration contains only two VAX 6000 computers, contact a Compaq services technician to move the TBK70 controller card to another BI backplane within the same VAX computer. | Moving the controller card changes the controller letter of the tape drive without changing the unit number (for example, MUA6 becomes MUB6).<br>**Note:** The tape drives can have the same unit number. |

#### 6.2.2.9 Node Allocation Classes and RAID Array 210 and 230 Devices

If you have RAID devices connected to StorageWorks RAID Array 210 or 230 subsystems, you might experience device-naming problems when running in a cluster environment if nonzero node allocation classes are used. In this case, the RAID devices will be named $*n*$DR*cu*, where *n* is the (nonzero) node allocation class, *c* is the controller letter, and *u* is the unit number.

If multiple nodes in the cluster have the same (nonzero) node allocation class and these same nodes have RAID controllers, then RAID devices that are distinct might be given the same name (for example, $1$DRA0). This problem can lead to data corruption.

To prevent such problems, use the DR_UNIT_BASE system parameter, which causes the DR devices to be numbered sequentially, starting with the DR_UNIT_BASE value that you specify. For example, if the node allocation class is $1, the controller letter is A, and you set DR_UNIT_BASE on one cluster member to 10, the first device name generated by the RAID controller will be $1$DRA10, followed by $1$DRA11, $1$DRA12, and so forth.

To ensure unique DR device names, set the DR_UNIT_BASE number on each cluster member so that the resulting device numbers do not overlap. For example, you can set DR_UNIT_BASE on three cluster members to 10, 20, and 30 respectively. As long as each cluster member has 10 or fewer devices, the DR device numbers will be unique.

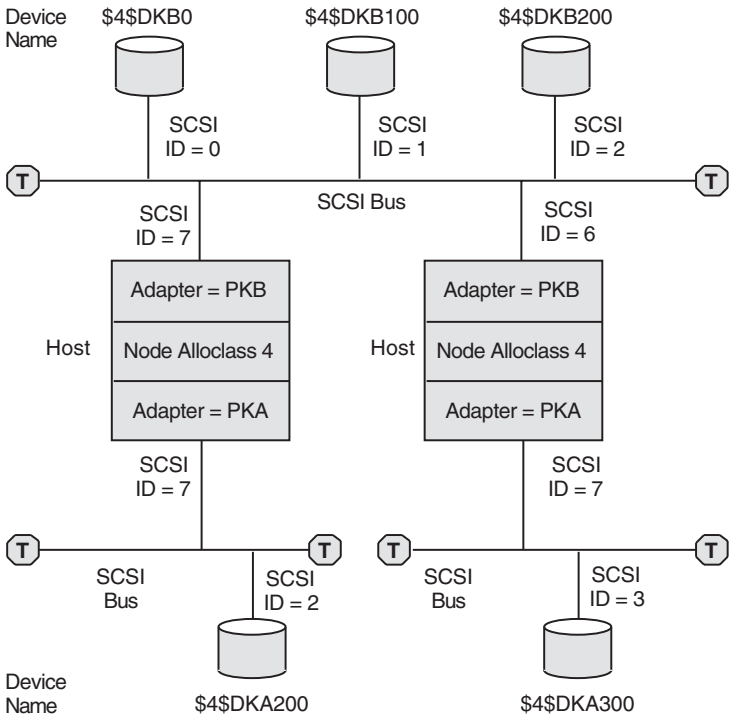
### 6.2.3 Reasons for Using Port Allocation Classes

When the node allocation class is nonzero, it becomes the device name prefix for all attached devices, whether the devices are on a shared interconnect or not. To ensure unique names within a cluster, it is necessary for the *ddcu* part of the disk device name (for example, DKB0) to be unique within an allocation class, even if the device is on a private bus.

This constraint is relatively easy to overcome for DIGITAL Storage Architecture (DSA) devices, because a system manager can select from a large unit number space to ensure uniqueness. The constraint is more difficult to manage for other device types, such as SCSI devices whose controller letter and unit number are determined by the hardware configuration.

For example, in the configuration shown in Figure 6–7, each system has a private SCSI bus with adapter letter A. To obtain unique names, the unit numbers must be different. This constrains the configuration to a maximum of 8 devices on the two buses (or 16 if wide addressing can be used on one or more of the buses). This can result in empty StorageWorks drive bays and in a reduction of the system's maximum storage capacity.

**Figure 6–7 SCSI Device Names Using a Node Allocation Class**




### 6.2.3.1 Constraint of the SCSI Controller Letter in Device Names

The SCSI device name is determined in part by the SCSI controller through which the device is accessed (for example, B in DKB*n*). Therefore, to ensure that each node uses the same name for each device, all SCSI controllers attached to a shared SCSI bus must have the same OpenVMS device name. In Figure 6–7, each host is attached to the shared SCSI bus by controller PKB.

This requirement can make configuring a shared SCSI bus difficult, because a system manager has little or no control over the assignment of SCSI controller device names. It is particularly difficult to match controller letters on different system types when one or more of the systems have:

- Built-in SCSI controllers that are not supported in SCSI clusters
- Long internal cables that make some controllers inappropriate for SCSI clusters

### 6.2.3.2 Constraints Removed by Port Allocation Classes

The **port allocation class** feature has two major benefits:

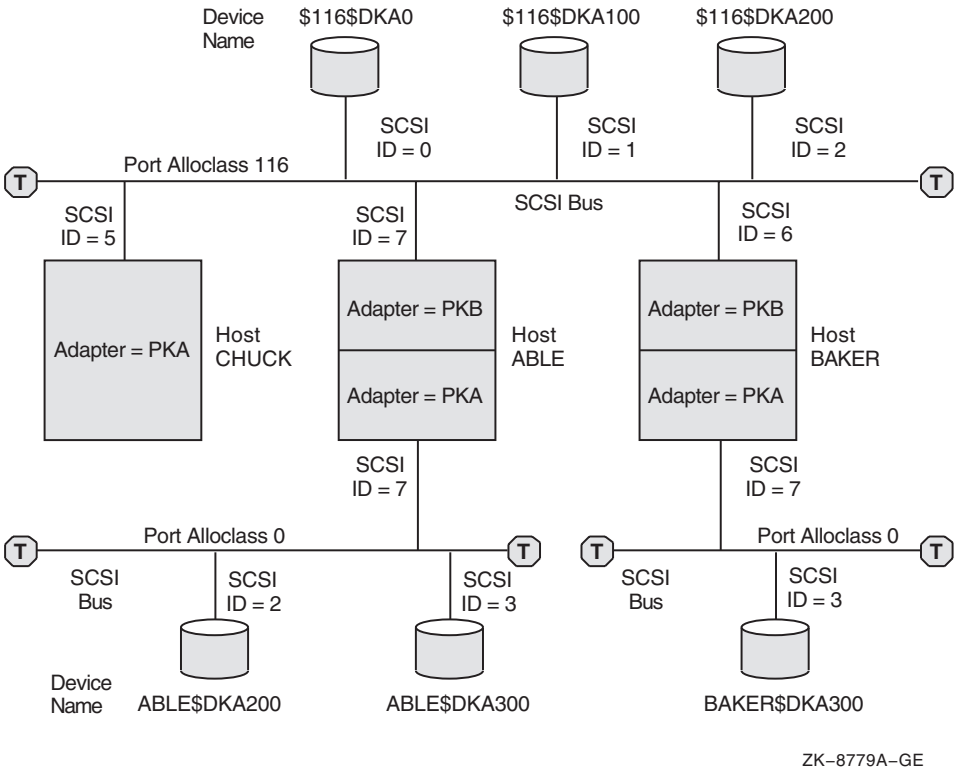
- A system manager can specify an allocation class value that is specific to a port rather than nodewide.
- When a port has a nonzero port allocation class, the controller letter in the device name that is accessed through that port is always the letter A.

Using port allocation classes for naming SCSI, IDE, floppy disk, and PCI RAID controller devices removes the configuration constraints described in Section 6.2.2.9, in Section 6.2.3, and in Section 6.2.3.1. You do not need to use the DR_UNIT_BASE system parameter recommended in Section 6.2.2.9. Furthermore, each bus can be given its own unique allocation class value, so the *ddcu* part of the disk device name (for example, DKB0) does not need to be unique across buses. Moreover, controllers with different device names can be attached to the same bus, because the disk device names no longer depend on the controller letter.

Figure 6–8 shows the same configuration as Figure 6–7, with two additions: a host named CHUCK and an additional disk attached to the lower left SCSI bus. Port allocation classes are used in the device names in this figure. A port allocation class of 116 is used for the SCSI interconnect that is shared, and port allocation class 0 is used for the SCSI interconnects that are not shared. By using port allocation classes in this configuration, you can do what was not allowed previously:

- Attach an adapter with a name (PKA) that differs from the name of the other adapters (PKB) attached to the shared SCSI interconnect, as long as that port has the same port allocation class (116 in this example).
- Use two disks with the same controller name and number (DKA300) because each disk is attached to a SCSI interconnect that is not shared.

**Figure 6–8 Device Names Using Port Allocation Classes**



### 6.2.4 Specifying Port Allocation Classes

A port allocation class is a designation for all ports attached to a single interconnect. It replaces the node allocation class in the device name.

The three types of port allocation classes are:

- Port allocation classes of 1 to 32767 for devices attached to a multihost interconnect or a single-host interconnect, if desired
- Port allocation class 0 for devices attached to a single-host interconnect
- Port allocation class -1 when no port allocation class is in effect

Each type has its own naming rules.

#### 6.2.4.1 Port Allocation Classes for Devices Attached to a Multi-Host Interconnect

The following rules pertain to port allocation classes for devices attached to a multihost interconnect:

1. The valid range of port allocation classes is 1 through 32767.
2. When using port allocation classes, the controller letter in the device name is always A, regardless of the actual controller letter. The $GETDVI item code DVI$_DISPLAY_DEVNAM displays the actual port name.

   Note that it is now more important to use fully specified names (for example, $101$DKA100 or ABLE$DKA100) rather than abbreviated names (such as DK100), because a system can have multiple DKA100 disks.
3. Each port allocation class must be unique within a cluster.
4. A port allocation class cannot duplicate the value of another node's tape or disk node allocation class.
5. Each node for which MSCP serves a device should have the same nonzero allocation class value.

Examples of device names that use this type of port allocation class are shown in Table 6–4.

**Table 6–4 Examples of Device Names with Port Allocation Classes 1-32767**

| Device Name | Description |
|---|---|
| $101$DKA0 | The port allocation class is 101; DK represents the disk device category, A is the controller name, and 0 is the unit number. |
| $147$DKA0 | The port allocation class is 147; DK represents the disk device category, A is the controller name, and 0 is the unit number. |

#### 6.2.4.2 Port Allocation Class 0 for Devices Attached to a Single-Host Interconnect

The following rules pertain to port allocation class 0 for devices attached to a single-host interconnect:

1. Port allocation class 0 does not become part of the device name. Instead, the name of the node to which the device is attached becomes the first part of the device name.
2. The controller letter in the device name remains the designation of the controller to which the device is attached. (It is not changed to A as it is for port allocation classes greater than zero.)

Examples of device names that use port allocation class 0 are shown in Table 6–5.

**Table 6–5 Examples of Device Names With Port Allocation Class 0**

| Device Name | Description |
|---|---|
| ABLE$DKD100 | ABLE is the name of the node to which the device is attached. D is the designation of the controller to which it is attached, not A as it is for port allocation classes with a nonzero class. The unit number of this device is 100. The port allocation class of $0$ is not included in the device name. |
| BAKER$DKC200 | BAKER is the name of the node to which the device is attached, C is the designation of the controller to which it is attached, and 200 is the unit number. The port allocation class of $0$ is not included in the device name. |

#### 6.2.4.3 Port Allocation Class -1

The designation of port allocation class -1 means that a port allocation class is not being used. Instead, a node allocation class is used. The controller letter remains its predefined designation. (It is assigned by OpenVMS, based on the system configuration. It is not affected by a node allocation class.)

#### 6.2.4.4 How to Implement Port Allocation Classes

Port allocation classes were introduced in OpenVMS Alpha Version 7.1 with support in OpenVMS VAX. VAX computers can serve disks connected to Alpha systems that use port allocation classes in their names.

To implement port allocation classes, you must do the following:

- Enable the use of port allocation classes.
- Assign one or more port allocation classes.
- At a minimum, reboot the nodes on the shared SCSI bus.

##### Enabling the Use of Port Allocation Classes

To enable the use of port allocation classes, you must set a new SYSGEN parameter DEVICE_NAMING to 1. The default setting for this parameter is zero. In addition, the SCSSYSTEMIDH system parameter must be set to zero. Check to make sure that it is.

##### Assigning Port Allocation Classes

You can assign one or more port allocation classes with the OpenVMS Cluster configuration procedure, CLUSTER_CONFIG.COM (or CLUSTER_CONFIG_LAN.COM).

If it is not possible to use CLUSTER_CONFIG.COM or CLUSTER_CONFIG_LAN.COM to assign port allocation classes (for example, if you are booting a private system disk into an existing cluster), you can use the new SYSBOOT SET/CLASS command.

The following example shows how to use the new SYSBOOT SET/CLASS command to assign an existing port allocation class of 152 to port PKB.

```
SYSBOOT> SET/CLASS PKB 152
```

The SYSINIT process ensures that this new name is used in successive boots.

To deassign a port allocation class, enter the port name without a class number. For example:

```
SYSBOOT> SET/CLASS PKB
```

The mapping of ports to allocation classes is stored in SYS$SYSTEM:SYS$DEVICES.DAT, a standard text file. You use the CLUSTER_CONFIG.COM (or CLUSTER_CONFIG_LAN.COM) command procedure or, in special cases, SYSBOOT to change SYS$DEVICES.DAT.

#### 6.2.4.5 Clusterwide Reboot Requirements for SCSI Interconnects

Changing a device's allocation class changes the device name. A clusterwide reboot ensures that all nodes see the device under its new name, which in turn means that the normal device and file locks remain consistent.

Rebooting an entire cluster when a device name changes is not mandatory. You may be able to reboot only the nodes that share the SCSI bus, as described in the following steps. The conditions under which you can do this and the results that follow are also described.

1. Dismount the devices whose names have changed from all nodes.

   This is not always possible. In particular, you cannot dismount a disk on nodes where it is the system disk. If the disk is not dismounted, a subsequent attempt to mount the same disk using the new device name will fail with the following error:

   ```
   %MOUNT-F-VOLALRMNT, another volume of same label already mounted
   ```

   Therefore, you must reboot any node that cannot dismount the disk.

2. Reboot all nodes connected to the SCSI bus.

   Before you reboot any of these nodes, make sure the disks on the SCSI bus are dismounted on the nodes not rebooting.

   **Note**

   OpenVMS ensures that a node cannot boot if the result is a SCSI bus with naming different from another node already accessing the same bus. (This check is independent of the dismount check in step 1.)

   After the nodes that are connected to the SCSI bus reboot, the device exists with its new name.

3. Mount the devices systemwide or clusterwide.

   If no other node has the disk mounted under the old name, you can mount the disk systemwide or clusterwide using its new name. The new device name will be seen on all nodes running compatible software, and these nodes can also mount the disk and access it normally.

   Nodes that have not rebooted still see the old device name as well as the new device name. However, the old device name cannot be used; the device, when accessed by the old name, is off line. The old name persists until the node reboots.

## 6.3 MSCP and TMSCP Served Disks and Tapes

The MSCP server and the TMSCP server make locally connected disks and tapes available to all cluster members. Locally connected disks and tapes are not automatically cluster accessible. Access to these devices is restricted to the local computer unless you explicitly set them up as cluster accessible using the MSCP server for disks or the TMSCP server for tapes.

### 6.3.1 Enabling Servers

To make a disk or tape accessible to all OpenVMS Cluster computers, the MSCP or TMSCP server must be:

- Loaded on the local computer, as described in Table 6–6
- Made functional by setting the MSCP and TMSCP system parameters, as described in Table 6–7

**Table 6–6 MSCP_LOAD and TMSCP_LOAD Parameter Settings**

| Parameter | Value | Meaning |
|---|---|---|
| MSCP_LOAD | 0 | Do not load the MSCP_SERVER. This is the default. |
| | 1 | Load the MSCP server with attributes specified by the MSCP_SERVE_ALL parameter using the default CPU load capacity. |
| | >1 | Load the MSCP server with attributes specified by the MSCP_SERVE_ALL parameter. Use the MSCP_LOAD value as the CPU load capacity. |
| TMSCP_LOAD | 0 | Do not load the TMSCP server and do not serve any tapes (default value). |
| | 1 | Load the TMSCP server and serve all available tapes, including all local tapes and all multihost tapes with a matching TAPE_ALLOCLASS value. |

Table 6–7 summarizes the system parameter values you can specify for MSCP_SERVE_ALL and TMSCP_SERVE_ALL to configure the MSCP and TMSCP servers. Initial values are determined by your responses when you execute the installation or upgrade procedure or when you execute the CLUSTER_CONFIG.COM command procedure described in Chapter 8 to set up your configuration.

Starting with OpenVMS Version 7.2, the serving types are implemented as a bit mask. To specify the type of serving your system will perform, locate the type you want in Table 6–7 and specify its value. For some systems, you may want to specify two serving types, such as serving the system disk and serving locally attached disks. To specify such a combination, add the values of each type, and specify the sum.

**Note**

In a mixed-version cluster that includes any systems running OpenVMS Version 7.1-*x* or earlier, serving all available disks is restricted to serving all disks whose allocation class matches the system's node allocation class (pre-Version 7.2 meaning). To specify this type of serving, use the value 9 (which sets bit 0 and bit 3).

**Table 6–7 MSCP_SERVE_ALL and TMSCP_SERVE_ALL Parameter Settings**

| Parameter | Bit | Value When Set | Meaning |
|---|---|---|---|
| MSCP_SERVE_ALL | 0 | 1 | Serve all available disks (locally attached and those connected to HS*x* and DSSI controllers). Disks with allocation classes that differ from the system's allocation class (set by the ALLOCLASS parameter) are also served if bit 3 is not set. |
| | 1 | 2 | Serve locally attached (non-HS*x* and non-DSSI) disks. The server does not monitor its I/O traffic and does not participate in load balancing. |
| | 2 | 4 | Serve the system disk. This is the default setting. This setting is important when other nodes in the cluster rely on this system being able to serve its system disk. This setting prevents obscure contention problems that can occur when a system attempts to complete I/O to a remote system disk whose system has failed. |
| | 3 | 8 | Restrict the serving specified by bit 0. All disks except those with allocation classes that differ from the system's allocation class (set by the ALLOCLASS parameter) are served.<br>This is pre-Version 7.2 behavior. If your cluster includes systems running Open 7.1-*x* or earlier, and you want to serve all available disks, you must specify 9, the result of setting this bit and bit 0. |
| TMSCP_SERVE_ALL | 0 | 1 | Serve all available tapes (locally attached and those connected to HS*x* and DSSI controllers). Tapes with allocation classes that differ from the system's allocation class (set by the ALLOCLASS parameter) are also served if bit 3 is not set. |
| | 1 | 2 | Serve locally attached (non-HS*x* and non-DSSI) tapes. |
| | 3 | 8 | Restrict the serving specified by bit 0. Serve all tapes except those with allocation classes that differ from the system's allocation class (set by the ALLOCLASS parameter).<br>This is pre-Version 7.2 behavior. If your cluster includes systems running OpenVMS Version 7.1-*x* or earlier, and you want to serve all available tapes, you must specify 9, the result of setting this bit and bit 0. |

Although the serving types are now implemented as a bit mask, the values of 0, 1, and 2, specified by bit 0 and bit 1, retain their original meanings. These values are shown in the following table:

| Value | Description |
|---|---|
| 0 | Do not serve any disks (tapes). This is the default. |
| 1 | Serve all available disks (tapes). |
| 2 | Serve only locally attached (non-HS*x* and non-DSSI) disks (tapes). |

#### 6.3.1.1 Serving the System Disk

Setting bit 2 to serve the system disk is important when other nodes in the cluster rely on this system being able to serve its system disk. This setting prevents obscure contention problems that can occur when a system attempts to complete I/O to a remote system disk whose system has failed.

The following sequence of events describes how a contention problem can occur if serving the system disk is disabled (that is, if bit 2 is not set):

- The MSCP_SERVE_ALL setting is changed to disable serving when the system reboots.
- The serving system crashes.
- The client system that was executing I/O to the serving system's system disk is holding locks on resources of that system disk.
- The client system starts mount verification.
- The serving system attempts to boot but cannot because of the locks held on its system disk by the client system.
- The client's mount verification process times out after a period of time set by the MVTIMEOUT system parameter, and the client system releases the locks. The time period could be several hours.
- The serving system is able to reboot.

#### 6.3.1.2 Setting the MSCP and TMSCP System Parameters

Use either of the following methods to set these system parameters:

- Specify appropriate values for these parameters in a computer's MODPARAMS.DAT file and then run AUTOGEN.
- Run the CLUSTER_CONFIG.COM or the CLUSTER_CONFIG_LAN.COM procedure, as appropriate, and choose the CHANGE option to perform these operations for disks and tapes.

With either method, the served devices become accessible when the serving computer reboots. Further, the servers automatically serve any suitable device that is added to the system later. For example, if new drives are attached to an HSC subsystem, the devices are dynamically configured.

**Note:** The SCSI retention command modifier is not supported by the TMSCP server. Retention operations should be performed from the node serving the tape.

## 6.4 MSCP I/O Load Balancing

MSCP I/O load balancing offers the following advantages:

- Faster I/O response
- Balanced work load among the members of an OpenVMS Cluster

Two types of MSCP I/O load balancing are provided by OpenVMS Cluster software: static and dynamic. Static load balancing occurs on both VAX and Alpha systems; dynamic load balancing occurs only on VAX systems. Both types of load balancing are based on the load capacity ratings of the server systems.

### 6.4.1 Load Capacity

The load capacity ratings for the VAX and Alpha systems are predetermined by Compaq. These ratings are used in the calculation of the available serving capacity for MSCP static and dynamic load balancing. You can override these default settings by specifying a different load capacity with the MSCP_LOAD parameter.

Note that the MSCP server load-capacity values (either the default value or the value you specify with MSCP_LOAD) are estimates used by the load-balancing feature. They cannot change the actual MSCP serving capacity of a system.

A system's MSCP serving capacity depends on many factors including its power, the performance of its LAN adapter, and the impact of other processing loads. The available serving capacity, which is calculated by each MSCP server as described in Section 6.4.3, is used solely to bias the selection process when a client system (for example, a satellite) chooses which server system to use when accessing a served disk.

### 6.4.2 Increasing the Load Capacity When FDDI is Used

When FDDI is used instead of Ethernet, the throughput is far greater. To take advantage of this greater throughput, Compaq recommends that you change the server's load-capacity default setting with the MSCP_LOAD parameter. Start with a multiplier of four. For example, the load-capacity rating of any Alpha system connected by FDDI to a disk can be set to 1360 I/O per second (4x340). Depending on your configuration and the software you are running, you may want to increase or decrease this value.

### 6.4.3 Available Serving Capacity

The load-capacity ratings are used by each MSCP server to calculate its available serving capacity.

The **available serving capacity** is calculated in the following way:

| Step | Calculation |
|---|---|
| **1** | Each MSCP server counts the read and write requests sent to it and periodically converts this value to requests per second. |
| **2** | Each MSCP server subtracts its requests per second from its load capacity to compute its available serving capacity. |

### 6.4.4 Static Load Balancing

MSCP servers periodically send their available serving capacities to the MSCP class driver (DUDRIVER). When a disk is mounted or one fails over, DUDRIVER assigns the server with the highest available serving capacity to it. (TMSCP servers do not perform this monitoring function.) This initial assignment is called static load balancing.

### 6.4.5 Dynamic Load Balancing (VAX Only)

Dynamic load balancing occurs only on VAX systems. MSCP server activity is checked every 5 seconds. If activity to any server is excessive, the serving load automatically shifts to other servers in the cluster.

### 6.4.6 Overriding MSCP I/O Load Balancing for Special Purposes

In some configurations, you may want to designate one or more systems in your cluster as the primary I/O servers and restrict I/O traffic on other systems. You can accomplish these goals by overriding the default load-capacity ratings used by the MSCP server. For example, if your cluster consists of two Alpha systems and one VAX 6000-400 system and you want to reduce the MSCP served I/O traffic to the VAX, you can assign a low MSCP_LOAD value, such as 50, to the VAX. Because the two Alpha systems each start with a load-capacity rating of 340 and the VAX now starts with a load-capacity rating of 50, the MSCP served satellites will direct most of the I/O traffic to the Alpha systems.

## 6.5 Managing Cluster Disks With the Mount Utility

For locally connected disks to be accessible to other nodes in the cluster, the MSCP server software must be loaded on the computer to which the disks are connected (see Section 6.3.1). Further, each disk must be mounted with the Mount utility, using the appropriate qualifier: /CLUSTER, /SYSTEM, or /GROUP. Mounting multiple disks can be automated with command procedures; a sample command procedure, MSCPMOUNT.COM, is provided in the SYS$EXAMPLES directory on your system.

The Mount utility also provides other qualifiers that determine whether a disk is automatically rebuilt during a remount operation. Different rebuilding techniques are recommended for data and system disks.

This section describes how to use the Mount utility for these purposes.

### 6.5.1 Mounting Cluster Disks

To mount disks that are to be shared among all computers, specify the MOUNT command as shown in the following table.

| IF... | THEN... |
|---|---|
| **At system startup** | |
| The disk is attached to a single system and is to be made available to all other nodes in the cluster. | Use MOUNT/CLUSTER *device-name* on the computer to which the disk is to be mounted. The disk is mounted on every computer that is active in the cluster at the time the command executes. First, the disk is mounted locally. Then, if the mount operation succeeds, the disk is mounted on other nodes in the cluster. |
| The computer has no disks directly attached to it. | Use MOUNT/SYSTEM *device-name* on the computer for each disk the computer needs to access. The disks can be attached to a single system or shared disks that are accessed by an HS*x* controller. Then, if the mount operation succeeds, the disk is mounted on the computer joining the cluster. |
| **When the system is running** | |
| You want to add a disk. | Use MOUNT/CLUSTER *device-name* on the computer to which the disk is to be mounted. The disk is mounted on every computer that is active in the cluster at the time the command executes. First, the disk is mounted locally. Then, if the mount operation succeeds, the disk is mounted on other nodes in the cluster. |

To ensure disks are mounted whenever possible, regardless of the sequence that systems in the cluster boot (or shut down), startup command procedures should

use MOUNT/CLUSTER and MOUNT/SYSTEM as described in the preceding table.

**Note:** Only system or group disks can be mounted across the cluster or on a subset of the cluster members. If you specify MOUNT/CLUSTER without the /SYSTEM or /GROUP qualifier, /SYSTEM is assumed. Also note that each cluster disk mounted with the /SYSTEM or /GROUP qualifier must have a unique volume label.

### 6.5.2 Examples of Mounting Shared Disks

Suppose you want all the computers in a three-member cluster to share a disk named COMPANYDOCS. To share the disk, one of the three computers can mount COMPANYDOCS using the MOUNT/CLUSTER command, as follows:

```
$ MOUNT/CLUSTER/NOASSIST $1$DUA4: COMPANYDOCS
```

If you want just two of the three computers to share the disk, those two computers must both mount the disk with the same MOUNT command, as follows:

```
$ MOUNT/SYSTEM/NOASSIST $1$DUA4: COMPANYDOCS
```

To mount the disk at startup time, include the MOUNT command either in a common command procedure that is invoked at startup time or in the computer-specific startup command file.

**Note:** The /NOASSIST qualifier is used in command procedures that are designed to make several attempts to mount disks. The disks may be temporarily offline or otherwise not available for mounting. If, after several attempts, the disk cannot be mounted, the procedure continues. The /ASSIST qualifier, which is the default, causes a command procedure to stop and query the operator if a disk cannot be mounted immediately.

### 6.5.3 Mounting Cluster Disks With Command Procedures

To configure cluster disks, you can create command procedures to mount them. You may want to include commands that mount cluster disks in a separate command procedure file that is invoked by a site-specific SYSTARTUP procedure. Depending on your cluster environment, you can set up your command procedure in either of the following ways:

- As a separate file specific to each computer in the cluster by making copies of the common procedure and storing them as separate files
- As a common computer-independent file on a shared disk

With either method, each computer can invoke the common procedure from the site-specific SYSTARTUP procedure.

**Example:** The MSCPMOUNT.COM file in the SYS$EXAMPLES directory on your system is a sample command procedure that contains commands typically used to mount cluster disks. The example includes comments explaining each phase of the procedure.

### 6.5.4 Disk Rebuild Operation

To minimize disk I/O operations (and thus improve performance) when files are created or extended, the OpenVMS file system maintains a cache of preallocated file headers and disk blocks.

If a disk is dismounted improperly—for example, if a system fails or is removed from a cluster without running SYS$SYSTEM:SHUTDOWN.COM—this preallocated space becomes temporarily unavailable. When the disk is remounted, MOUNT scans the disk to recover the space. This is called a **disk rebuild operation**.

### 6.5.5 Rebuilding Cluster Disks

On a nonclustered computer, the MOUNT scan operation for recovering preallocated space merely prolongs the boot process. In an OpenVMS Cluster system, however, this operation can degrade response time for all user processes in the cluster. While the scan is in progress on a particular disk, most activity on that disk is blocked.

**Note:** User processes that attempt to read or write to files on the disk can experience delays of several minutes or longer, especially if the disk contains a large number of files or has many users.

Because the rebuild operation can delay access to disks during the startup of any OpenVMS Cluster computer, Compaq recommends that procedures for mounting cluster disks use the /NOREBUILD qualifier. When MOUNT/NOREBUILD is specified, disks are not scanned to recover lost space, and users experience minimal delays while computers are mounting disks.

**Reference:** Section 6.5.6 provides information about rebuilding system disks. Section 9.5.1 provides more information about disk rebuilds and system-disk throughput techniques.

### 6.5.6 Rebuilding System Disks

Rebuilding system disks is especially critical because most system activity requires access to a system disk. When a system disk rebuild is in progress, very little activity is possible on any computer that uses that disk.

Unlike other disks, the system disk is automatically mounted early in the boot sequence. If a rebuild is necessary, and if the value of the system parameter ACP_REBLDSYSD is 1, the system disk is rebuilt during the boot sequence. (The default setting of 1 for the ACP_REBLDSYSD system parameter specifies that the system disk should be rebuilt.) Exceptions are as follows:

| Setting | Comments |
|---|---|
| ACP_REBLDSYSD parameter should be set to 0 on satellites. | This setting prevents satellites from rebuilding a system disk when it is mounted early in the boot sequence and eliminates delays caused by such a rebuild when satellites join the cluster. |
| ACP_REBLDSYSD should be set to the default value of 1 on boot servers, and procedures that mount disks on the boot servers should use the /REBUILD qualifier. | While these measures can make boot server rebooting more noticeable, they ensure that system disk space is available after an unexpected shutdown. |

Once the cluster is up and running, system managers can submit a batch procedure that executes SET VOLUME/REBUILD commands to recover lost disk space. Such procedures can run at a time when users would not be

inconvenienced by the blocked access to disks (for example, between midnight and 6 a.m. each day). Because the SET VOLUME/REBUILD command determines whether a rebuild is needed, the procedures can execute the command for each disk that is usually mounted.

**Suggestion:** The procedures run more quickly and cause less delay in disk access if they are executed on:

- Powerful computers
- Computers that have direct access to the volume to be rebuilt

Moreover, several such procedures, each of which rebuilds a different set of disks, can be executed simultaneously.

**Caution:** If either or both of the following conditions are true when mounting disks, it is essential to run a procedure with SET VOLUME/REBUILD commands on a regular basis to rebuild the disks:

- Disks are mounted with the MOUNT/NOREBUILD command.
- The ACP_REBLDSYSD system parameter is set to 0.

Failure to rebuild disk volumes can result in a loss of free space and in subsequent failures of applications to create or extend files.

## 6.6 Shadowing Disks Across an OpenVMS Cluster

Volume shadowing (sometimes referred to as disk mirroring) achieves high data availability by duplicating data on multiple disks. If one disk fails, the remaining disk or disks can continue to service application and user I/O requests.

### 6.6.1 Purpose

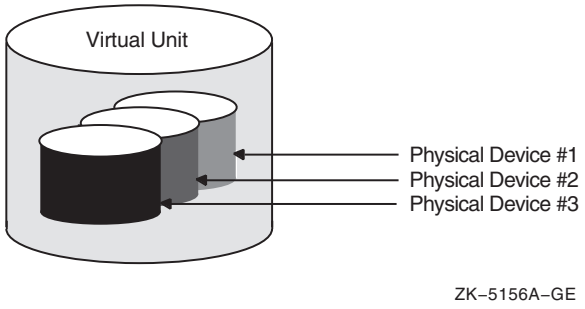
Volume Shadowing for OpenVMS software provides data availability across the full range of OpenVMS configurations—from single nodes to large OpenVMS Cluster systems—so you can provide data availabililty where you need it most.

Volume Shadowing for OpenVMS software is an implementation of RAID 1 (redundant arrays of independent disks) technology. Volume Shadowing for OpenVMS prevents a disk device failure from interrupting system and application operations. By duplicating data on multiple disks, volume shadowing transparently prevents your storage subsystems from becoming a single point of failure because of media deterioration, communication path failure, or controller or device failure.

### 6.6.2 Shadow Sets

You can mount one, two, or three compatible disk volumes to form a **shadow set**, as shown in Figure 6–9. Each disk in the shadow set is known as a shadow set **member**. Volume Shadowing for OpenVMS logically binds the shadow set devices together and represents them as a single virtual device called a **virtual unit**. This means that the multiple members of the shadow set, represented by the virtual unit, appear to operating systems and users as a single, highly available disk.

**Figure 6–9 Shadow Set With Three Members**




### 6.6.3 I/O Capabilities

Applications and users read and write data to and from a shadow set using the same commands and program language syntax and semantics that are used for nonshadowed I/O operations. System managers manage and monitor shadow sets using the same commands and utilities they use for nonshadowed disks. The only difference is that access is through the virtual unit, not to individual devices.

**Reference:** *Volume Shadowing for OpenVMS* describes the shadowing product capabilities in detail.

### 6.6.4 Supported Devices

For a single workstation or a large data center, valid shadowing configurations include:

- All MSCP compliant DSA drives
- All DSSI devices
- All StorageWorks SCSI disks and controllers, and some third-party SCSI devices that implement READL (read long) and WRITEL (write long) commands and use the SCSI disk driver (DKDRIVER)

  **Restriction:** SCSI disks that do not support READL and WRITEL are restricted because these disks do not support the shadowing data repair (disk bad-block errors) capability. Thus, using unsupported SCSI disks can cause members to be removed from the shadow set.

You can shadow data disks and system disks. Thus, a system disk need not be a single point of failure for any system that boots from that disk. System disk shadowing becomes especially important for OpenVMS Cluster systems that use a *common* system disk from which multiple computers boot.

Volume Shadowing for OpenVMS does not support the shadowing of quorum disks. This is because volume shadowing makes use of the OpenVMS distributed lock manager, and the quorum disk must be utilized before locking is enabled.

There are no restrictions on the location of shadow set members beyond the valid disk configurations defined in the Volume Shadowing for OpenVMS *Software Product Description* (SPD 27.29.*xx*).

### 6.6.5 Shadow Set Limits

You can mount a maximum of 500 shadow sets (each having one, two, or three members) in a standalone or OpenVMS Cluster system. The number of shadow sets supported is independent of controller and device types. The shadow sets can be mounted as public or private volumes.

For any changes to these limits, consult the Volume Shadowing for OpenVMS *Software Product Description* (SPD 27.29.*xx*).

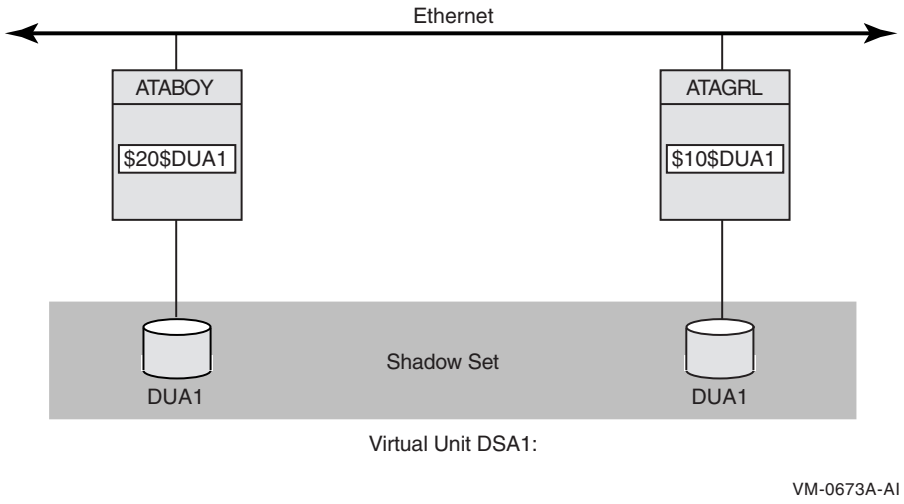
### 6.6.6 Distributing Shadowed Disks

The controller-independent design of shadowing allows you to manage shadow sets regardless of their controller connection or location in the OpenVMS Cluster system and helps provide improved data availability and very flexible configurations.

For clusterwide shadowing, members can be located anywhere in an OpenVMS Cluster system and served by MSCP servers across any supported OpenVMS Cluster interconnect, including the CI, Ethernet, DSSI, and FDDI. For example, OpenVMS Cluster systems using FDDI can be up to 40 kilometers apart, which further increases the availability and disaster tolerance of a system.

Figure 6–10 shows how shadow set member units are on line to local controllers located on different nodes. In the figure, a disk volume is local to each of the nodes ATABOY and ATAGRL. The MSCP server provides access to the shadow set members over the Ethernet. Even though the disk volumes are local to different nodes, the disks are members of the same shadow set. A member unit that is local to one node can be accessed by the remote node over the MSCP server.

**Figure 6–10 Shadow Sets Accessed Through the MSCP Server**




For shadow sets that are mounted on an OpenVMS Cluster system, mounting or dismounting a shadow set on one node in the cluster does not affect applications or user functions executing on other nodes in the system. For example, you can dismount the virtual unit from one node in an OpenVMS Cluster system and leave the shadow set operational on the remaining nodes on which it is mounted.

Other shadowing notes:

- If an individual disk volume is already mounted as a member of an active shadow set, the disk volume cannot be mounted as a standalone disk on another node.
- System disks can be shadowed. All nodes booting from shadowed system disks must:
  - Have a Volume Shadowing for OpenVMS license.
  - Specify the same physical member of the system disk shadow set as the boot device.
  - Set shadowing system parameters to enable shadowing and specify the system disk virtual unit number.
  - Mount additional physical members into the system disk shadow set early in the SYSTARUP_VMS.COM command procedure.
  - Mount the disks to be used in the shadow set.

# 7

# Setting Up and Managing Cluster Queues

This chapter discusses queuing topics specific to OpenVMS Cluster systems. Because queues in an OpenVMS Cluster system are established and controlled with the same commands used to manage queues on a standalone computer, the discussions in this chapter assume some knowledge of queue management on a standalone system, as described in the *OpenVMS System Manager's Manual*.

**Note:** See the *OpenVMS System Manager's Manual* for information about queuing compatibility.

## 7.1 Introduction

Users can submit jobs to any queue in the OpenVMS Cluster system, regardless of the processor on which the job will actually execute. Generic queues can balance the work load among the available processors.

The system manager can use one or several queue managers to manage batch and print queues for an entire OpenVMS Cluster system. Although a single queue manager is sufficient for most systems, multiple queue managers can be useful for distributing the batch and print work load across nodes in the cluster.

**Note:** OpenVMS Cluster systems that include both VAX and Alpha computers must use the queue manager described in this chapter.

## 7.2 Controlling Queue Availability

Once the batch and print queue characteristics are set up, the system manager can rely on the distributed queue manager to make queues available across the cluster.

The distributed queue manager prevents the queuing system from being affected when a node enters or leaves the cluster during cluster transitions. The following table describes how the distributed queue manager works.

| WHEN... | THEN... | Comments |
|---|---|---|
| The node on which the queue manager is running leaves the OpenVMS Cluster system. | The queue manager automatically fails over to another node. | This failover occurs transparently to users on the system. |
| Nodes are added to the cluster. | The queue manager automatically serves the new nodes. | The system manager does not need to enter a command explicitly to start queuing on the new node. |
| The OpenVMS Cluster system reboots. | The queuing system automatically restarts by default. | Thus, you do not have to include commands in your startup command procedure for queuing. |

| WHEN... | THEN... | Comments |
|---|---|---|
| | The operating system automatically restores the queuing system with the parameters defined in the queuing database. | This is because when you start the queuing system, the characteristics you define are stored in a queue database. |

To control queues, the queue manager maintains a clusterwide queue database that stores information about queues and jobs. Whether you use one or several queue managers, only one queue database is shared across the cluster. Keeping the information for all processes in one database allows jobs submitted from any computer to execute on any queue (provided that the necessary mass storage devices are accessible).

## 7.3 Starting a Queue Manager and Creating the Queue Database

You start up a queue manager using the START/QUEUE/MANAGER command as you would on a standalone computer. However, in an OpenVMS Cluster system, you can also provide a failover list and a unique name for the queue manager. The /NEW_VERSION qualifier creates a new queue database.

The following command example shows how to start a queue manager:

```
$ START/QUEUE/MANAGER/NEW_VERSION/ON=(GEM,STONE,*)
```

The following table explains the components of this sample command.

| Command | Function |
|---|---|
| START/QUEUE/MANAGER | Creates a single, clusterwide queue manager named SYS$QUEUE_MANAGER. |
| /NEW_VERSION | Creates a new queue database in SYS$COMMON:[SYSEXE] that consists of the following three files:<br>• QMAN$MASTER.DAT (master file)<br>• SYS$QUEUE_MANAGER.QMAN$QUEUES (queue file)<br>• SYS$QUEUE_MANAGER.QMAN$JOURNAL (journal file)<br>**Rule:** Use the /NEW_VERSION qualifier only on the first invocation of the queue manager or if you want to create a new queue database. |
| /ON=*(node-list)*<br>[optional] | Specifies an ordered list of nodes that can claim the queue manager if the node running the queue manager should exit the cluster. In the example:<br>• The queue manager process starts on node GEM.<br>• If the queue manager is running on node GEM and GEM leaves the cluster, the queue manager fails over to node STONE.<br>• The asterisk wildcard (*) is specified as the last node in the node list to indicate that any remaining, unlisted nodes can start the queue manager in any order.<br>**Rules:** Complete node names are required; you cannot specify the asterisk wildcard character as part of a node name.<br>If you want to exclude certain nodes from being eligible to run the queue manager, do not use the asterisk wildcard character in the node list. |

| Command | Function |
|---|---|
| /NAME_OF_MANAGER [optional] | Allows you to assign a unique name to the queue manager. Unique queue manager names are necessary if you run multiple queue managers. For example, using the /NAME_OF_MANAGER qualifier causes queue and journal files to be created using the queue manager name instead of the default name SYS$QUEUE_MANAGER. For example, adding the /NAME_OF_MANAGER=PRINT_MANAGER qualifier command creates these files:<br><br>QMAN$MASTER.DAT<br>PRINT_MANAGER.QMAN$QUEUES<br>PRINT_MANAGER.QMAN$JOURNAL |

**Rules for OpenVMS Cluster systems with multiple system disks:**

- Specify the locations of both the master file and the queue and journal files for systems that do not boot from the system disk where the files are located.

  **Reference:** If you want to locate the queue database files on other devices or directories, refer to the *OpenVMS System Manager's Manual* for instructions.

- Specify a device and directory that is accessible across the OpenVMS Cluster.
- Define the device and directory identically in the SYS$COMMON:SYLOGICALS.COM startup command procedure on every node.

## 7.4 Starting Additional Queue Managers

Running multiple queue managers balances the work load by distributing batch and print jobs across the cluster. For example, you might create separate queue managers for batch and print queues in clusters with CPU or memory shortages. This allows the batch queue manager to run on one node while the print queue manager runs on a different node.

### 7.4.1 Command Format

To start additional queue managers, include the /ADD and /NAME_OF_MANAGER qualifiers on the START/QUEUE/MANAGER command. Do not specify the /NEW_VERSION qualifier. For example:

```
$ START/QUEUE/MANAGER/ADD/NAME_OF_MANAGER=BATCH_MANAGER
```

### 7.4.2 Database Files

Multiple queue managers share one QMAN$MASTER.DAT master file, but an additional queue file and journal file is created for each queue manager. The additional files are named in the following format, respectively:

- *name_of_manager*.QMAN$QUEUES
- *name_of_manager*.QMAN$JOURNAL

By default, the queue database and its files are located in SYS$COMMON:[SYSEXE]. If you want to relocate the queue database files, refer to the instructions in Section 7.6.

## 7.5 Stopping the Queuing System

When you enter the STOP/QUEUE/MANAGER/CLUSTER command, the queue manager remains stopped, and requests for queuing are denied until you enter the START/QUEUE/MANAGER command (without the /NEW_VERSION qualifier).

The following command shows how to stop a queue manager named PRINT_MANAGER:

```
$ STOP/QUEUE/MANAGER/CLUSTER/NAME_OF_MANAGER=PRINT_MANAGER
```

**Rule:** You must include the /CLUSTER qualifier on the command line whether or not the queue manager is running on an OpenVMS Cluster system. If you omit the /CLUSTER qualifier, the command stops all queues on the default node without stopping the queue manager. (This has the same effect as entering the STOP/QUEUE/ON_NODE command.)

## 7.6 Moving Queue Database Files

The files in the queue database can be relocated from the default location of SYS$COMMON:[SYSEXE] to any disk that is mounted clusterwide or that is accessible to the computers participating in the clusterwide queue scheme. For example, you can enhance system performance by locating the database on a shared disk that has a low level of activity.

### 7.6.1 Location Guidelines

The master file QMAN$MASTER can be in a location separate from the queue and journal files, but the queue and journal files must be kept together in the same directory. The queue and journal files for one queue manager can be separate from those of other queue managers.

The directory you specify must be available to all nodes in the cluster. If the directory specification is a concealed logical name, it must be defined identically in the SYS$COMMON:SYLOGICALS.COM startup command procedure on every node in the cluster.

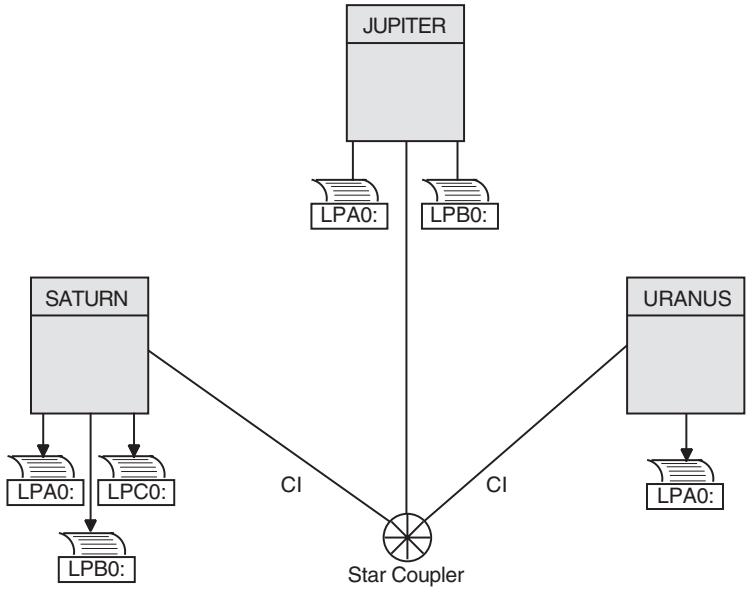
**Reference:** The *OpenVMS System Manager's Manual* contains complete information about creating or relocating the queue database files. See also Section 7.12 for a sample common procedure that sets up an OpenVMS Cluster batch and print system.

## 7.7 Setting Up Print Queues

To establish print queues, you must determine the type of queue configuration that best suits your OpenVMS Cluster system. You have several alternatives that depend on the number and type of print devices you have on each computer and on how you want print jobs to be processed. For example, you need to decide:

- Which print queues you want to establish on each computer
- Whether to set up any clusterwide generic queues to distribute print job processing across the cluster
- Whether to set up autostart queues for availability or improved startup time

**Figure 7–1 Sample Printer Configuration**




ZK–1631–GE

Once you determine the appropriate strategy for your cluster, you can create your queues. Figure 7–1 shows the printer configuration for a cluster consisting of the active computers JUPITR, SATURN, and URANUS.

### 7.7.1 Creating a Queue

You set up OpenVMS Cluster print queues using the same method that you would use for a standalone computer. However, in an OpenVMS Cluster system, you must provide a unique name for each queue you create.

### 7.7.2 Command Format

You create and name a print queue by specifying the INITIALIZE/QUEUE command at the DCL prompt in the following format:

INITIALIZE/QUEUE/ON=node-name::device[/START][/NAME_OF_MANAGER=*name-of-manager*] queue-name

| Qualifier | Description |
|---|---|
| /ON | Specifies the computer and printer to which the queue is assigned. If you specify the /START qualifier, the queue is started. |
| /NAME_OF_MANAGER | If you are running multiple queue managers, you should also specify the queue manager with the qualifier. |

### 7.7.3 Ensuring Queue Availability

You can also use the autostart feature to simplify startup and ensure high availability of execution queues in an OpenVMS Cluster. If the node on which the autostart queue is running leaves the OpenVMS Cluster, the queue automatically fails over to the next available node on which autostart is enabled. Autostart is particularly useful on LAT queues. Because LAT printers are usually shared among users of multiple systems or in OpenVMS Cluster systems, many users are affected if a LAT queue is unavailable.

**Format for creating autostart queues:**

Create an autostart queue with a list of nodes on which the queue can run by specifying the DCL command INITIALIZE/QUEUE in the following format:

```
INITIALIZE/QUEUE/AUTOSTART_ON=(node-name::device:,node-name::device:, . . . ) queue-name
```

When you use the /AUTOSTART_ON qualifier, you must initially activate the queue for autostart, either by specifying the /START qualifier with the INITIALIZE /QUEUE command or by entering a START/QUEUE command. However, the queue cannot begin processing jobs until the ENABLE AUTOSTART /QUEUES command is entered for a node on which the queue can run. Generic queues cannot be autostart queues.

**Rules:** Generic queues cannot be autostart queues. Note that you cannot specify both /ON and /AUTOSTART_ON.

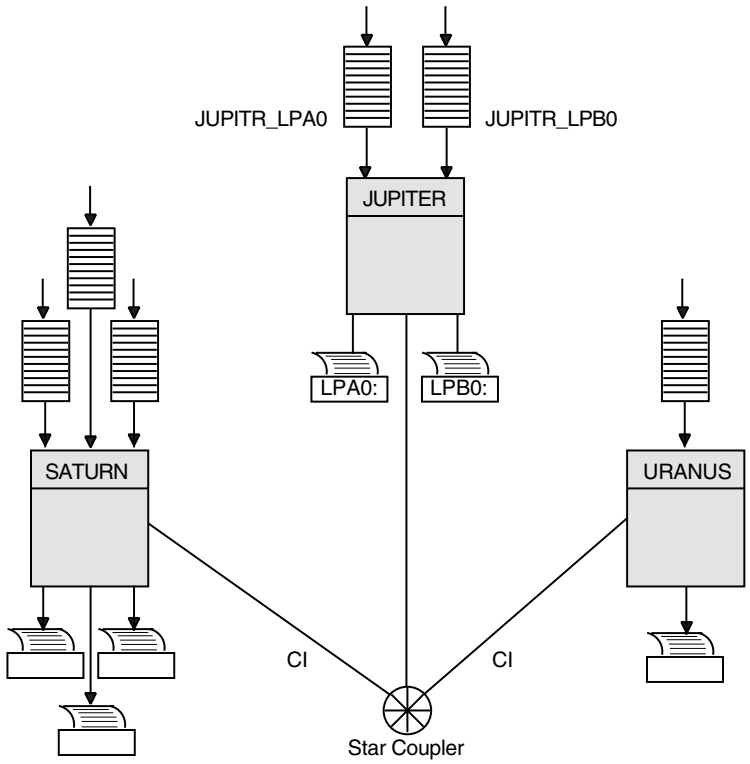
**Reference:** Refer to Section 7.13 for information about setting the time at which autostart is disabled.

### 7.7.4 Examples

The following commands make the local print queue assignments for JUPITR shown in Figure 7–2 and start the queues:

```
$ INITIALIZE/QUEUE/ON=JUPITR::LPA0/START/NAME_OF_MANAGER=PRINT_MANAGER JUPITR_LPA0
$ INITIALIZE/QUEUE/ON=JUPITR::LPB0/START/NAME_OF_MANAGER=PRINT_MANAGER JUPITR_LPB0
```

**Figure 7–2 Print Queue Configuration**




## 7.8 Setting Up Clusterwide Generic Print Queues

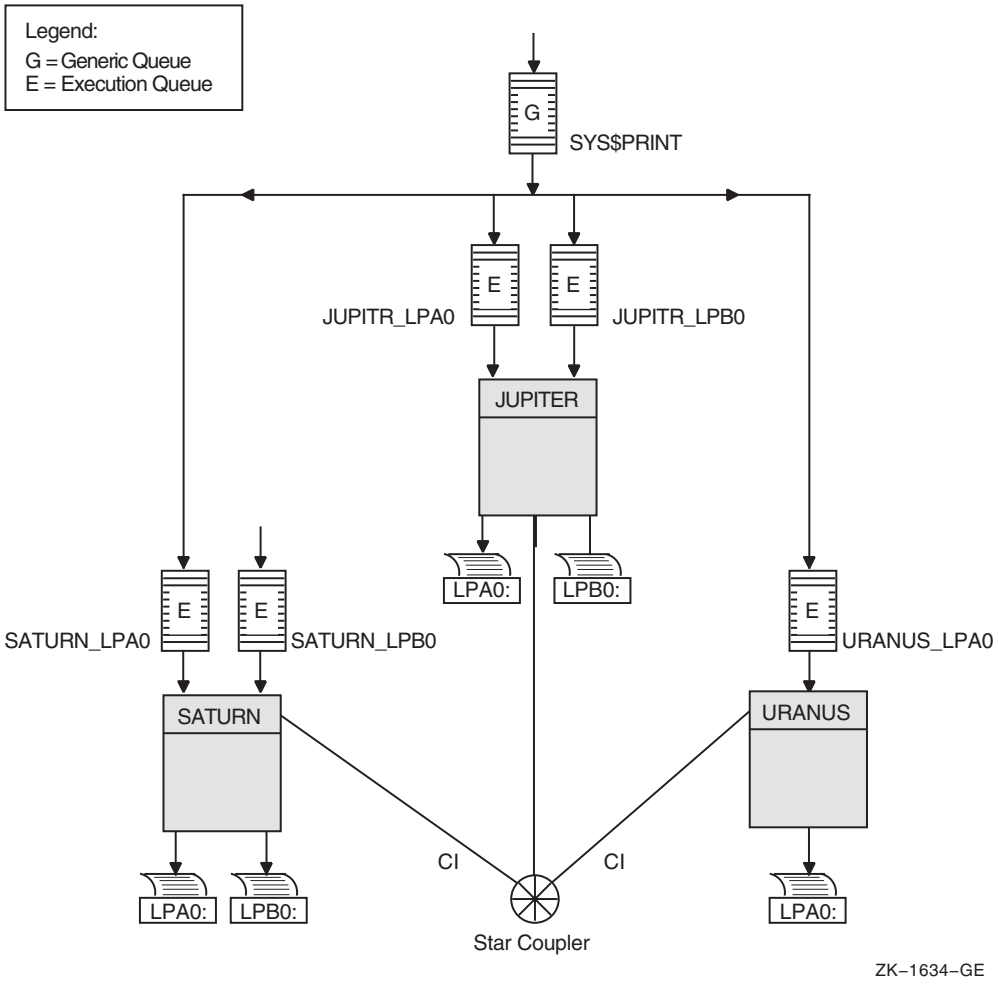
The clusterwide queue database enables you to establish generic queues that function throughout the cluster. Jobs queued to clusterwide generic queues are placed in any assigned print queue that is available, regardless of its location in the cluster. However, the file queued for printing must be accessible to the computer to which the printer is connected.

### 7.8.1 Sample Configuration

Figure 7–3 illustrates a clusterwide generic print queue in which the queues for all LPA0 printers in the cluster are assigned to a clusterwide generic queue named SYS$PRINT.

A clusterwide generic print queue needs to be initialized and started only once. The most efficient way to start your queues is to create a common command procedure that is executed by each OpenVMS Cluster computer (see Section 7.12.3).

**Figure 7–3 Clusterwide Generic Print Queue Configuration**




## 7.8.2 Command Example

The following command initializes and starts the clusterwide generic queue SYS$PRINT:

```
$ INITIALIZE/QUEUE/GENERIC=(JUPITR_LPA0,SATURN_LPA0,URANUS_LPA0)/START SYS$PRINT
```

Jobs queued to SYS$PRINT are placed in whichever assigned print queue is available. Thus, in this example, a print job from JUPITR that is queued to SYS$PRINT can be queued to JUPITR_LPA0, SATURN_LPA0, or URANUS_LPA0.

# 7.9 Setting Up Execution Batch Queues

Generally, you set up execution batch queues on each OpenVMS Cluster computer using the same procedures you use for a standalone computer. For more detailed information about how to do this, see the *OpenVMS System Manager's Manual*.

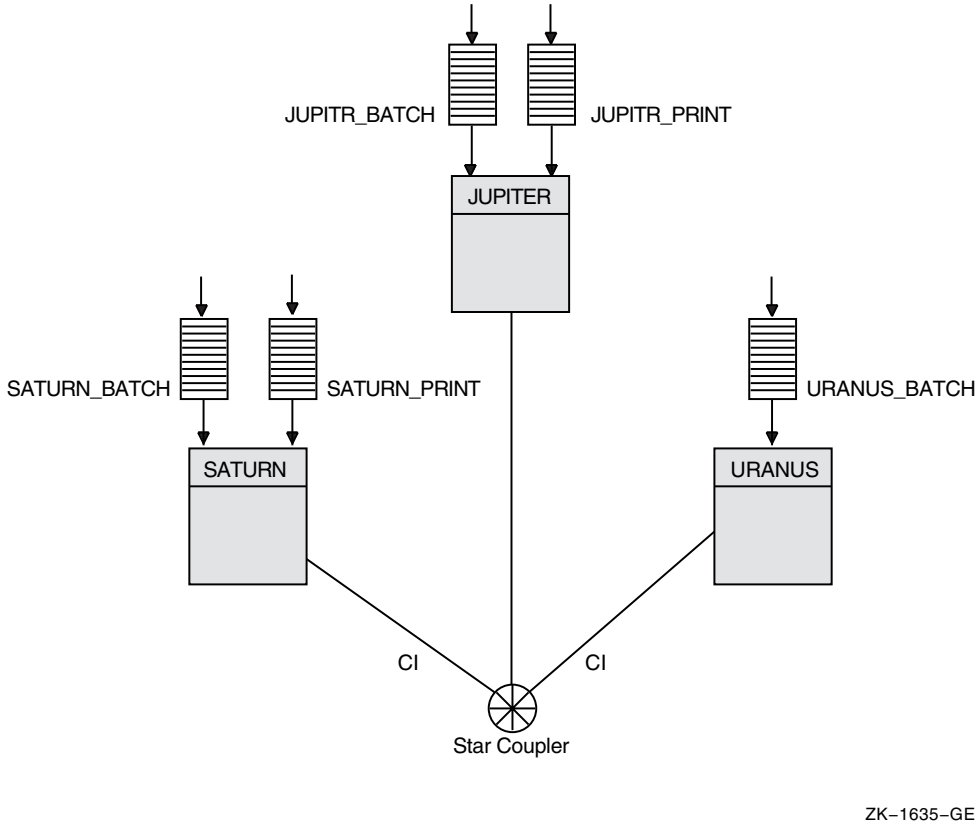
### 7.9.1 Before You Begin

Before you establish batch queues, you should decide which type of queue configuration best suits your cluster. As system manager, you are responsible for setting up batch queues to maintain efficient batch job processing on the cluster. For example, you should do the following:

- Determine what type of processing will be performed on each computer.
- Set up local batch queues that conform to these processing needs.
- Decide whether to set up any clusterwide generic queues that will distribute batch job processing across the cluster.
- Decide whether to use autostart queues for startup simplicity.

Once you determine the strategy that best suits your needs, you can create a command procedure to set up your queues. Figure 7–4 shows a batch queue configuration for a cluster consisting of computers JUPITR, SATURN, and URANUS.

**Figure 7–4 Sample Batch Queue Configuration**



### 7.9.2 Batch Command Format

You create a batch queue with a unique name by specifying the DCL command INITIALIZE/QUEUE/BATCH in the following format:

INITIALIZE/QUEUE/BATCH/ON=node::[/START][/NAME_OF_MANAGER=*name-of-manager*] queue-name

| Qualifier | Description |
|---|---|
| /ON | Specifies the computer on which the batch queue runs. |
| /START | Starts the queue. |
| /NAME_OF_MANAGER | Specifies the name of the queue manager if you are running multiple queue managers. |

### 7.9.3 Autostart Command Format

You can initialize and start an autostart batch queue by specifying the DCL command INITIALIZE/QUEUE/BATCH. Use the following command format:

INITIALIZE/QUEUE/BATCH/AUTOSTART_ON=node::queue-name

When you use the /AUTOSTART_ON qualifier, you must initially activate the queue for autostart, either by specifying the /START qualifier with the INITIALIZE/QUEUE command or by entering a START/QUEUE command. However, the queue cannot begin processing jobs until the ENABLE AUTOSTART /QUEUES command is entered for a node on which the queue can run.

**Rule:** Generic queues cannot be autostart queues. Note that you cannot specify both /ON and /AUTOSTART_ON.

### 7.9.4 Examples

The following commands make the local batch queue assignments for JUPITR, SATURN, and URANUS shown in Figure 7–4:

```
$ INITIALIZE/QUEUE/BATCH/ON=JUPITR::/START/NAME_OF_MANAGER=BATCH_QUEUE JUPITR_BATCH
$ INITIALIZE/QUEUE/BATCH/ON=SATURN::/START/NAME_OF_MANAGER=BATCH_QUEUE SATURN_BATCH
$ INITIALIZE/QUEUE/BATCH/ON=URANUS::/START/NAME_OF_MANAGER=BATCH_QUEUE URANUS_BATCH
```

Because batch jobs on each OpenVMS Cluster computer are queued to SYS$BATCH by default, you should consider defining a logical name to establish this queue as a clusterwide generic batch queue that distributes batch job processing throughout the cluster (see Example 7–2). Note, however, that you should do this only if you have a common-environment cluster.

## 7.10 Setting Up Clusterwide Generic Batch Queues

In an OpenVMS Cluster system, you can distribute batch processing among computers to balance the use of processing resources. You can achieve this workload distribution by assigning local batch queues to one or more clusterwide generic batch queues. These generic batch queues control batch processing across the cluster by placing batch jobs in assigned batch queues that are available. You can create a clusterwide generic batch queue as shown in Example 7–2.

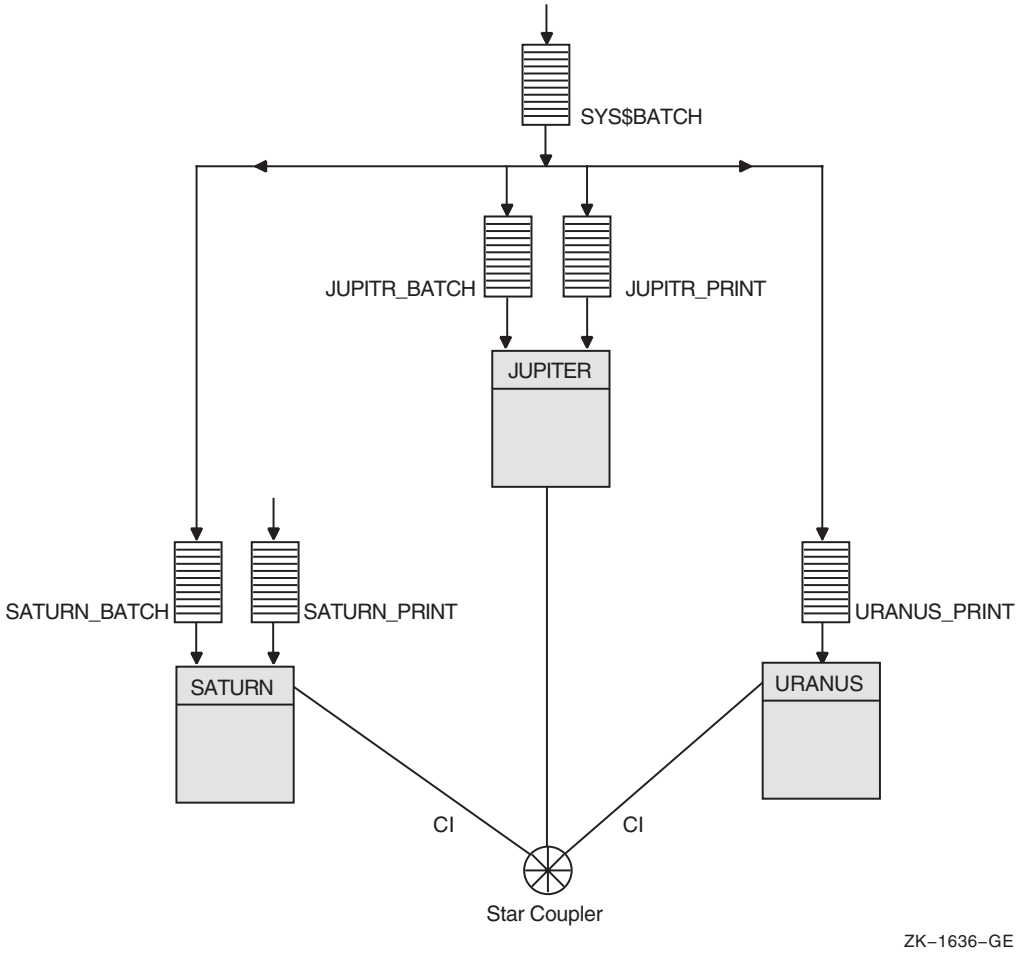
A clusterwide generic batch queue needs to be initialized and started only once. The most efficient way to perform these operations is to create a common command procedure that is executed by each OpenVMS Cluster computer (see Example 7–2).

### 7.10.1 Sample Configuration

In Figure 7–5, batch queues from each OpenVMS Cluster computer are assigned to a clusterwide generic batch queue named SYS$BATCH. Users can submit a job to a specific queue (for example, JUPITR_BATCH or SATURN_BATCH), or, if they have no special preference, they can submit it by default to the clusterwide generic queue SYS$BATCH. The generic queue in turn places the job in an available assigned queue in the cluster.

If more than one assigned queue is available, the operating system selects the queue that minimizes the ratio (executing jobs/job limit) for all assigned queues.

**Figure 7–5 Clusterwide Generic Batch Queue Configuration**




## 7.11 Starting Local Batch Queues

Normally, you use local batch execution queues during startup to run batch jobs to start layered products. For this reason, these queues must be started before the ENABLE AUTOSTART command is executed, as shown in the command procedure in Example 7–1.

### 7.11.1 Startup Command Procedure

Start the local batch execution queue in each node's startup command procedure SYSTARTUP_VMS.COM. If you use a common startup command procedure, add commands similar to the following to your procedure:

```
$  SUBMIT/PRIORITY=255/NOIDENT/NOLOG/QUEUE=node_BATCH LAYERED_PRODUCT.COM
$  START/QUEUE node_BATCH
$  DEFINE/SYSTEM/EXECUTIVE SYS$BATCH node_BATCH
```

Submitting the startup command procedure LAYERED_PRODUCT.COM as a high-priority batch job before the queue starts ensures that the job is executed immediately, regardless of the job limit on the queue. If the queue is started before the command procedure was submitted, the queue might reach its job limit by scheduling user batch jobs, and the startup job would have to wait.

## 7.12 Using a Common Command Procedure

Once you have created queues, you must start them to begin processing batch and print jobs. In addition, you must make sure the queues are started each time the system reboots, by enabling autostart for autostart queues or by entering START/QUEUE commands for nonautostart queues. To do so, create a command procedure containing the necessary commands.

### 7.12.1 Command Procedure

You can create a common command procedure named, for example, QSTARTUP.COM, and store it on a shared disk. With this method, each node can share the same copy of the common QSTARTUP.COM procedure. Each node invokes the common QSTARTUP.COM procedure from the common version of SYSTARTUP. You can also include the commands to start queues in the common SYSTARTUP file instead of in a separate QSTARTUP.COM file.

### 7.12.2 Examples

Example 7–1 shows commands used to create OpenVMS Cluster queues.

**Example 7–1 Sample Commands for Creating OpenVMS Cluster Queues**

```
$
❶
$  DEFINE/FORM LN_FORM 10 /WIDTH=80 /STOCK=DEFAULT /TRUNCATE
$  DEFINE/CHARACTERISTIC 2ND_FLOOR 2
   .
   .
   .
❷
$  INITIALIZE/QUEUE/AUTOSTART_ON=(JUPITR::LPA0:)/START JUPITR_PRINT
$  INITIALIZE/QUEUE/AUTOSTART_ON=(SATURN::LPA0:)/START SATURN_PRINT
$  INITIALIZE/QUEUE/AUTOSTART_ON=(URANUS::LPA0:)/START URANUS_PRINT
   .
   .
   .
❸
$  INITIALIZE/QUEUE/BATCH/START/ON=JUPITR:: JUPITR_BATCH
$  INITIALIZE/QUEUE/BATCH/START/ON=SATURN:: SATURN_BATCH
$  INITIALIZE/QUEUE/BATCH/START/ON=URANUS:: URANUS_BATCH
   .
   .
   .
❹
$  INITIALIZE/QUEUE/START -
_$ /AUTOSTART_ON=(JUPITR::LTA1:,SATURN::LTA1,URANUS::LTA1) -
_$ /PROCESSOR=LATSYM /FORM_MOUNTED=LN_FORM -
_$ /RETAIN=ERROR /DEFAULT=(NOBURST,FLAG=ONE,NOTRAILER) -
_$ /RECORD_BLOCKING LN03$PRINT
$
$  INITIALIZE/QUEUE/START -
_$ /AUTOSTART_ON=(JUPITR::LTA2:,SATURN::LTA2,URANUS::LTA2) -
_$ /PROCESSOR=LATSYM /RETAIN=ERROR -
_$ /DEFAULT=(NOBURST,FLAG=ONE,NOTRAILER) /RECORD_BLOCKING -
_$ /CHARACTERISTIC=2ND_FLOOR LA210$PRINT
$
❺
$  ENABLE AUTOSTART/QUEUES/ON=SATURN
$  ENABLE AUTOSTART/QUEUES/ON=JUPITR
$  ENABLE AUTOSTART/QUEUES/ON=URANUS
❻
$  INITIALIZE/QUEUE/START SYS$PRINT -
_$ /GENERIC=(JUPITR_PRINT,SATURN_PRINT,URANUS_PRINT)
$
❼
$  INITIALIZE/QUEUE/BATCH/START SYS$BATCH -
_$ /GENERIC=(JUPITR_BATCH,SATURN_BATCH,URANUS_BATCH)
$
```

Following are descriptions of each command or group of commands in Example 7–1.

| Command | Description |
|---|---|
| ❶ | Define all printer forms and characteristics. |
| ❷ | Initialize local print queues. In the example, these queues are autostart queues and are started automatically when the node executes the ENABLE AUTOSTART/QUEUES command. Although the /START qualifier is specified to activate the autostart queues, they do not begin processing jobs until autostart is enabled.<br><br>To enable autostart each time the system reboots, add the ENABLE AUTOSTART/QUEUES command to your queue startup command procedure, as shown in Example 7–2. |
| ❸ | Initialize and start local batch queues on all nodes, including satellite nodes. In this example, the local batch queues are not autostart queues. |
| ❹ | Initialize queues for remote LAT printers. In the example, these queues are autostart queues and are set up to run on one of three nodes. The queues are started on the first of those three nodes to execute the ENABLE AUTOSTART command.<br><br>You must establish the logical devices LTA1 and LTA2 in the LAT startup command procedure LAT$SYSTARTUP.COM on each node on which the autostart queue can run. For more information, see the description of editing LAT$SYSTARTUP.COM in the *OpenVMS System Manager's Manual*.<br><br>Although the /START qualifier is specified to activate these autostart queues, they will not begin processing jobs until autostart is enabled. |
| ❺ | Enable autostart to start the autostart queues automatically. In the example, autostart is enabled on node SATURN first, so the queue manager starts the autostart queues that are set up to run on one of several nodes. |
| ❻ | Initialize and start the generic output queue SYS$PRINT. This is a nonautostart queue (generic queues cannot be autostart queues). However, generic queues are not stopped automatically when a system is shut down, so you do not need to restart the queue each time a node reboots. |
| ❼ | Initialize and start the generic batch queue SYS$BATCH. Because this is a generic queue, it is not stopped when the node shuts down. Therefore, you do not need to restart the queue each time a node reboots. |

### 7.12.3 Example

Example 7–2 illustrates the use of a common QSTARTUP command procedure on a shared disk.

**Example 7–2 Common Procedure to Start OpenVMS Cluster Queues**

```
$!
$! QSTARTUP.COM -- Common procedure to set up cluster queues
$!
$!
❶
$ NODE = F$GETSYI("NODENAME")
$!
$! Determine the node-specific subroutine
$!
$ IF (NODE .NES. "JUPITR") .AND. (NODE .NES. "SATURN") .AND. (NODE .NES. "URANUS")
$    THEN
$       GOSUB SATELLITE_STARTUP
$    ELSE
❷
$!
$!      Configure remote LAT devices.
$!
$       SET TERMINAL LTA1: /PERM /DEVICE=LN03 /WIDTH=255 /PAGE=60 -
              /LOWERCASE /NOBROAD
$       SET TERMINAL LTA2: /PERM /DEVICE=LA210 /WIDTH=255 /PAGE=66 -
                 /NOBROAD
$       SET DEVICE LTA1: /SPOOLED=(LN03$PRINT,SYS$SYSDEVICE:)
$       SET DEVICE LTA2: /SPOOLED=(LA210$PRINT,SYS$SYSDEVICE:)
❸
$       START/QUEUE/BATCH 'NODE'_BATCH
$       GOSUB 'NODE'_STARTUP
$    ENDIF
$ GOTO ENDING
$!
$! Node-specific subroutines start here
$!
❹
$ SATELLITE_STARTUP:
$!
$! Start a batch queue for satellites.
$!
$ START/QUEUE/BATCH 'NODE'_BATCH
$ RETURN
$!
❺
$JUPITR_STARTUP:
$!
$! Node-specific startup for JUPITR::
$! Setup local devices and start nonautostart queues here
$!
$ SET PRINTER/PAGE=66 LPA0:
$ RETURN
```

(continued on next page)

**Example 7–2 (Cont.) Common Procedure to Start OpenVMS Cluster Queues**

```
$!
$SATURN_STARTUP:
$!
$! Node-specific startup for SATURN::
$! Setup local devices and start nonautostart queues here
$!
   .
   .
   .
$ RETURN
$!
$URANUS_STARTUP:
$!
$! Node-specific startup for URANUS::
$! Setup local devices and start nonautostart queues here
$!
   .
   .
   .
$ RETURN
$!
$ENDING:
❻
$! Enable autostart to start all autostart queues
$!
$ ENABLE AUTOSTART/QUEUES
$ EXIT
```

Following are descriptions of each phase of the common QSTARTUP.COM command procedure in Example 7–2.

| Command | Description |
|---|---|
| ❶ | Determine the name of the node executing the procedure. |
| ❷ | On all large nodes, set up remote devices connected by the LAT. The queues for these devices are autostart queues and are started automatically when the ENABLE AUTOSTART/QUEUES command is executed at the end of this procedure.<br>In the example, these autostart queues were set up to run on one of three nodes. The queues start when the first of those nodes executes the ENABLE AUTOSTART/QUEUES command. The queue remains running as long as one of those nodes is running and has autostart enabled. |
| ❸ | On large nodes, start the local batch queue. In the example, the local batch queues are nonautostart queues and must be started explicitly with START/QUEUE commands. |
| ❹ | On satellite nodes, start the local batch queue. |
| ❺ | Each node executes its own subroutine. On node JUPITR, set up the line printer device LPA0:. The queue for this device is an autostart queue and is started automatically when the ENABLE AUTOSTART/QUEUES command is executed. |
| ❻ | Enable autostart to start all autostart queues. |

## 7.13 Disabling Autostart During Shutdown

By default, the shutdown procedure disables autostart at the beginning of the shutdown sequence. Autostart is disabled to allow autostart queues with failover lists to fail over to another node. Autostart also prevents any autostart queue running on another node in the cluster to fail over to the node being shut down.

### 7.13.1 Options

You can change the time at which autostart is disabled in the shutdown sequence in one of two ways:

| Option | Description |
|---|---|
| **1** | Define the logical name SHUTDOWN$DISABLE_AUTOSTART as follows:<br>`$ DEFINE/SYSTEM/EXECUTIVE SHUTDOWN$DISABLE_AUTOSTART` *`number-of-minutes`*<br>Set the value of *number-of-minutes* to the number of minutes before shutdown when autostart is to be disabled. You can add this logical name definition to SYLOGICALS.COM. The value of *number-of-minutes* is the default value for the node. If this number is greater than the number of minutes specified for the entire shutdown sequence, autostart is disabled at the beginning of the sequence. |
| **2** | Specify the DISABLE_AUTOSTART *number-of-minutes* option during the shutdown procedure. (The value you specify for *number-of-minutes* overrides the value specified for the SHUTDOWN$DISABLE_AUTOSTART logical name.) |

**Reference:** See the *OpenVMS System Manager's Manual* for more information about changing the time at which autostart is disabled during the shutdown sequence.

# 8

# Configuring an OpenVMS Cluster System

This chapter provides an overview of the cluster configuration command procedures and describes the preconfiguration tasks required before running either command procedure. Then it describes each major function of the command procedures and the postconfiguration tasks, including running AUTOGEN.COM.

## 8.1 Overview of the Cluster Configuration Procedures

Two similar command procedures are provided for configuring and reconfiguring an OpenVMS Cluster system: CLUSTER_CONFIG_LAN.COM and CLUSTER_CONFIG.COM. The choice depends on whether you use the LANCP utility or DECnet for satellite booting in your cluster. CLUSTER_CONFIG_LAN.COM provides satellite booting services with the LANCP utility; CLUSTER_CONFIG.COM provides satellite booting sevices with DECnet. See Section 4.5 for the factors to consider when choosing a satellite booting service.

These configuration procedures automate most of the tasks required to configure an OpenVMS Cluster system. When you invoke CLUSTER_CONFIG_LAN.COM or CLUSTER_CONFIG.COM, the following configuration options are displayed:

- Add a computer to the cluster
- Remove a computer from the cluster
- Change a computer's characteristics
- Create a duplicate system disk
- Make a directory structure for a new root on a system disk
- Delete a root from a system disk

By selecting the appropriate option, you can configure the cluster easily and reliably without invoking any OpenVMS utilities directly. Table 8–1 summarizes the functions that the configuration procedures perform for each configuration option.

The phrase **cluster configuration command procedure**, when used in this chapter, refers to both CLUSTER_CONFIG_LAN.COM and CLUSTER_CONFIG.COM. The questions of the two configuration procedures are identical except where they pertain to LANCP and DECnet.

**Note:** For help on any question in these command procedures, type a question mark (?) at the question.

**Table 8–1 Summary of Cluster Configuration Functions**

| Option | Functions Performed |
|---|---|
| ADD | Enables a node as a cluster member:<br>• Establishes the new computer's root directory on a cluster common system disk and generates the computer's system parameter files (ALPHAVMSSYS.PAR for Alpha systems or VAXVMSSYS.PAR for VAX systems), and MODPARAMS.DAT in its SYS$SPECIFIC:[SYSEXE] directory.<br>• Generates the new computer's page and swap files (PAGEFILE.SYS and SWAPFILE.SYS).<br>• Sets up a cluster quorum disk (optional).<br>• Sets disk allocation class values, or port allocation class values (Alpha only), or both, with the ALLOCLASS parameter for the new computer, if the computer is being added as a disk server. If the computer is being added as a tape server, sets a tape allocation class value with the TAPE_ALLOCLASS parameter.<br>**Note:** ALLOCLASS must be set to a value greater than zero if you are configuring an Alpha computer on a shared SCSI bus and you are not using a port allocation class.<br>• Generates an initial (temporary) startup procedure for the new computer. This initial procedure:<br>– Runs NETCONFIG.COM to configure the network.<br>– Runs AUTOGEN to set appropriate system parameter values for the computer.<br>– Reboots the computer with normal startup procedures.<br>• If the new computer is a satellite node, the configuration procedure updates:<br>– Network databases for the computer on which the configuration procedure is executed to add the new computer.<br>– SYS$MANAGER:NETNODE_UPDATE.COM command procedure on the local computer (as described in Section 10.4.2). |
| REMOVE | Disables a node as a cluster member:<br>• Deletes another computer's root directory and its contents from the local computer's system disk. If the computer being removed is a satellite, the cluster configuration command procedure updates SYS$MANAGER:NETNODE_UPDATE.COM on the local computer.<br>• Updates the permanent and volatile remote node network databases on the local computer.<br>• Removes the quorum disk. |

(continued on next page)

**Table 8–1 (Cont.) Summary of Cluster Configuration Functions**

| Option | Functions Performed |
|---|---|
| CHANGE | Displays the CHANGE menu and prompts for appropriate information to:<br>• Enable or disable the local computer as a disk server<br>• Enable or disable the local computer as a boot server<br>• Enable or disable the Ethernet or FDDI LAN for cluster communications on the local computer<br>• Enable or disable a quorum disk on the local computer<br>• Change a satellite's Ethernet or FDDI hardware address<br>• Enable or disable the local computer as a tape server<br>• Change the local computer's ALLOCLASS or TAPE_ALLOCLASS value<br>• Change the local computer's shared SCSI port allocation class value<br>• Enable or disable MEMORY CHANNEL for node-to-node cluster communications on the local computer |
| CREATE | Duplicates the local computer's system disk and removes all system roots from the new disk. |
| MAKE | Creates a directory structure for a new root on a system disk. |
| DELETE | Deletes a root from a system disk. |

### 8.1.1 Before Configuring the System

Before invoking either the CLUSTER_CONFIG_LAN.COM or the CLUSTER_CONFIG.COM procedure to configure an OpenVMS Cluster system, perform the tasks described in Table 8–2.

**Table 8–2 Preconfiguration Tasks**

| Task | Procedure |
|---|---|
| Determine whether the computer uses DECdtm. | When you add a computer to or remove a computer from a cluster that uses DECdtm services, there are a number of tasks you need to do in order to ensure the integrity of your data.<br>**Reference:** See the chapter about DECdtm services in the *OpenVMS System Manager's Manual* for step-by-step instructions on setting up DECdtm in an OpenVMS Cluster system.<br>If you are not sure whether your cluster uses DECdtm services, enter this command sequence:<br>`$ SET PROCESS /PRIVILEGES=SYSPRV`<br>`$ RUN SYS$SYSTEM:LMCP`<br>`LMCP> SHOW LOG`<br>If your cluster does not use DECdtm services, the SHOW LOG command will display a "file not found" error message. If your cluster uses DECdtm services, it displays a list of the files that DECdtm uses to store information about transactions. |

(continued on next page)

**Table 8–2 (Cont.) Preconfiguration Tasks**

| Task | Procedure |
|---|---|
| Ensure the network software providing the satellite booting service is up and running and all computers are connected to the LAN. | For nodes that will use the LANCP utility for satellite booting, run the LANCP utility and enter the LANCP command LIST DEVICE/MOPDLL to display a list of LAN devices on the system:<br>`$ RUN SYS$SYSTEM:LANCP`<br>`LANCP> LIST DEVICE/MOPDLL`<br>For nodes running DECnet for OpenVMS, enter the DCL command SHOW NETWORK to determine whether the network is up and running:<br>`$ SHOW NETWORK`<br>`VAX/VMS Network status for local node 63.452 VIVID on 5-NOV-1994`<br>`This is a nonrouting node, and does not have any network information.`<br>`The designated router for VIVID is node 63.1021 SATURN.`<br>This example shows that the node VIVID is running DECnet for OpenVMS. If DECnet has not been started, the message "SHOW-I-NONET, Network Unavailable" is displayed.<br>For nodes running DECnet–Plus, refer to *DECnet for OpenVMS Network Management Utilities* for information about determining whether the DECnet–Plus network is up and running. |
| Select MOP and disk servers. | Every OpenVMS Cluster configured with satellite nodes must include at least one Maintenance Operations Protocol (MOP) and disk server. When possible, select multiple computers as MOP and disk servers. Multiple servers give better availability, and they distribute the work load across more LAN adapters.<br>Follow these guidelines when selecting MOP and disk servers:<br>• Ensure that MOP servers have direct access to the system disk.<br>• Ensure that disk servers have direct access to the storage that they are serving.<br>• Choose the most powerful computers in the cluster. Low-powered computers can become overloaded when serving many busy satellites or when many satellites boot simultaneously. Note, however, that two or more moderately powered servers may provide better performance than a single high-powered server.<br>• If you have several computers of roughly comparable power, it is reasonable to use them all as boot servers. This arrangement gives optimal load balancing. In addition, if one computer fails or is shut down, others remain available to serve satellites.<br>• After compute power, the most important factor in selecting a server is the speed of its LAN adapter. Servers should be equipped with the highest-bandwidth LAN adapters in the cluster. |
| Make sure you are logged in to a privileged account. | Log in to a privileged account.<br>**Rules:** If you are adding a satellite, you must be logged into the system manager's account on a boot server. Note that the process privileges SYSPRV, OPER, CMKRNL, BYPASS, and NETMBX are required, because the procedure performs privileged system operations. |
| Coordinate cluster common files. | If your configuration has two or more system disks, follow the instructions in Chapter 5 to coordinate the cluster common files. |
| Optionally, disable broadcast messages to your terminal. | While adding and removing computers, many such messages are generated. To disable the messages, you can enter the DCL command REPLY/DISABLE=(NETWORK, CLUSTER). See also Section 10.6 for more information about controlling OPCOM messages. |

(continued on next page)

**Table 8–2 (Cont.) Preconfiguration Tasks**

| Task | Procedure |
|---|---|
| Predetermine answers to the questions asked by the cluster configuration procedure. | Table 8–3 describes the data requested by the cluster configuration command procedures. |

### 8.1.2 Data Requested by the Cluster Configuration Procedures

The following table describes the questions asked by the cluster configuration command procedures and describes how you might answer them. The table is supplied here so that you can determine answers to the questions before you invoke the procedure.

Because many of the questions are configuration specific, Table 8–3 lists the questions according to configuration type, and not in the order they are asked.

**Table 8–3 Data Requested by CLUSTER_CONFIG_LAN.COM and CLUSTER_CONFIG.COM**

| Information Required | How to Specify or Obtain |
|---|---|
| **For all configurations** | |
| Device name of cluster system disk on which root directories will be created | Press Return to accept the default device name which is the translation of the SYS$SYSDEVICE: logical name, or specify a logical name that points to the common system disk. |
| Computer's root directory name on cluster system disk | Press Return to accept the procedure-supplied default, or specify a name in the form SYS*x*:<br>• For computers with direct access to the system disk, *x* is a hexadecimal digit in the range of 1 through 9 or A through D (for example, SYS1 or SYSA).<br>• For satellites, *x* must be in the range of 10 through FFFF. |
| Workstation windowing system | System manager specifies. Workstation software must be installed before workstation satellites are added. If it is not, the procedure indicates that fact. |

(continued on next page)

**Table 8–3 (Cont.) Data Requested by CLUSTER_CONFIG_LAN.COM and CLUSTER_CONFIG.COM**

| Information Required | How to Specify or Obtain |
|---|---|
| **For all configurations** | |
| Location and sizes of page and swap files | This information is requested only when you add a computer to the cluster. Press Return to accept the default size and location. (The default sizes displayed in brackets by the procedure are minimum values. The default location is the device name of the cluster system disk.)<br><br>If your configuration includes satellite nodes, you may realize a performance improvement by locating satellite page and swap files on a satellite's local disk, if such a disk is available. The potential for performance improvement depends on the configuration of your OpenVMS Cluster system disk and network.<br><br>To set up page and swap files on a satellite's local disk, the cluster configuration procedure creates a command procedure called SATELLITE_PAGE.COM in the satellite's [SYS*n*.SYSEXE] directory on the boot server's system disk. The SATELLITE_PAGE.COM procedure performs the following functions:<br><br>• Mounts the satellite's local disk with a volume label that is unique in the cluster in the format *node-name*_SCSSYSTEMID.<br><br>**Reference:** Refer to Section 8.6.5 for information about altering the volume label.<br><br>• Installs the page and swap files on the satellite's local disk.<br><br>**Note:** For page and swap disks that are shadowed, you must edit the MOUNT and INIT commands in SATELLITE_PAGE.COM to the appropriate syntax for mounting any specialized "local" disks (that is, host-based shadowing disks (DS*xxx*), or host-based RAID disks (DP*xxxx*), or DECram disks (MDA*xxxx*)) on the newly added node. CLUSTER_CONFIG(_LAN).COM does not create the MOUNT and INIT commands required for SHADOW, RAID, or DECram disks.<br><br>**Note:** To relocate the satellite's page and swap files (for example, from the satellite's local disk to the boot server's system disk, or the reverse) or to change file sizes:<br><br>1. Create new PAGE and SWAP files on a shared device, as shown:<br><br>`$ MCR SYSGEN CREATE device:[dir] PAGEFILE.SYS/SIZE=block-count`<br><br>**Note:** If page and swap files will be created for a shadow set, you must edit SATELLITE_PAGE accordingly.<br><br>2. Rename the SYS$SPECIFIC:[SYSEXE]PAGEFILE.SYS and SWAPFILE.SYS files to PAGEFILE.TMP and SWAPFILE.TMP.<br><br>3. Reboot, and then delete the .TMP files.<br><br>4. Modify the SYS$MANAGER:SYPAGSWPFILES.COM procedure to load the files. |
| Value for local computer's allocation class (ALLOCLASS or TAPE_ALLOCLASS) parameter. | The ALLOCLASS parameter can be used for a node allocation class or, on Alpha computers, a port allocation class. Refer to Section 6.2.1 for complete information about specifying allocation classes. |
| Physical device name of quorum disk | System manager specifies. |

(continued on next page)

**Table 8–3 (Cont.) Data Requested by CLUSTER_CONFIG_LAN.COM and CLUSTER_CONFIG.COM**

| Information Required | How to Specify or Obtain |
|---|---|
| **For systems running DECnet for OpenVMS** | |
| Computer's DECnet node address for Phase IV | For the DECnet node address, you obtain this information as follows:<br>• If you are adding a computer, the network manager supplies the address.<br>• If you are removing a computer, use the SHOW NETWORK command (as shown in Table 8–2). |
| Computer's DECnet node name | Network manager supplies. The name must be from 1 to 6 alphanumeric characters and *cannot* include dollar signs ( $ ) or underscores ( _ ). |
| **For systems running DECnet–Plus** | |
| Computer's DECnet node address for Phase IV (if you need Phase IV compatibility) | For the DECnet node address, you obtain this information as follows:<br>• If you are adding a computer, the network manager supplies the address.<br>• If you are removing a computer, use the SHOW NETWORK command (as shown in Table 8–2). |
| Node's DECnet full name | Determine the full name with the help of your network manager. Enter a string comprised of:<br>• The namespace name, ending with a colon (:). This is optional.<br>• The root directory, designated by a period (.).<br>• Zero or more hierarchical directories, designated by a character string followed by a period (.).<br>• The simple name, a character string that, combined with the directory names, uniquely identifies the node. For example:<br>.SALES.NETWORKS.MYNODE<br>MEGA:.INDIANA.JONES<br>COLUMBUS:.FLATWORLD |
| SCS node name for this node | Enter the OpenVMS Cluster node name, which is a string of 6 or fewer alphanumeric characters. |
| DECnet synonym | Press Return to define a DECnet synonym, which is a short name for the node's full name. Otherwise, enter N. |
| Synonym name for this node | Enter a string of 6 or fewer alphanumeric characters. By default, it is the first 6 characters of the last simple name in the full name. For example:<br>†Full name: BIGBANG:.GALAXY.NOVA.BLACKHOLE<br>Synonym: BLACKH<br>**Note:** The node synonym does not need to be the same as the OpenVMS Cluster node name. |
| MOP service client name for this node | Enter the name for the node's MOP service client when the node is configured as a boot server. By default, it is the OpenVMS Cluster node name (for example, the SCS node name). This name does not need to be the same as the OpenVMS Cluster node name. |

†DECnet–Plus full-name functionality is VAX specific.

(continued on next page)

**Table 8–3 (Cont.) Data Requested by CLUSTER_CONFIG_LAN.COM and CLUSTER_CONFIG.COM**

| Information Required | How to Specify or Obtain |
|---|---|
| **For systems running TCP/IP or the LANCP Utility for satellite booting, or both** | |
| Computer's SCS node name (SCSNODE) and SCS system ID (SCSSYSTEMID) | These prompts are described in Section 4.2.3. If a system is running TCP/IP, the procedure does not ask for a TCP/IP host name because a cluster node name (SCSNODE) does not have to match a TCP/IP host name. The TCP/IP host name might be longer than six characters, whereas the SCSNODE name must be no more than six characters. Note that if the system is running both DECnet and IP, then the procedure uses the DECnet defaults. |
| **For LAN configurations** | |
| Cluster group number and password | This information is requested only when the CHANGE option is chosen. See Section 2.5 for information about assigning cluster group numbers and passwords. |
| Satellite's LAN hardware address | Address has the form *xx-xx-xx-xx-xx-xx*. You must include the hyphens when you specify a hardware address. Proceed as follows:<br>• ‡On Alpha systems, enter the following command at the satellite's console:<br>`>>> SHOW NETWORK`<br>Note that you can also use the SHOW CONFIG command.<br>• †On MicroVAX II and VAXstation II satellite nodes. When the DECnet for OpenVMS network is running on a boot server, enter the following commands at the satellite's console:<br>`>>> B/100 XQA0`<br>`Bootfile: READ_ADDR`<br>• †On MicroVAX 2000 and VAXstation 2000 satellite nodes. When the DECnet for OpenVMS network is running on a boot server, enter the following commands at successive console-mode prompts:<br>`>>> T 53`<br>`2 ?>>> 3`<br>`>>> B/100 ESA0`<br>`Bootfile: READ_ADDR`<br>If the second prompt appears as 3 ?>>>, press Return.<br>• †On MicroVAX 3*xxx* and 4*xxx* series satellite nodes, enter the following command at the satellite's console:<br>`>>> SHOW ETHERNET` |

†DECnet–Plus full-name functionality is VAX specific.
‡Alpha specific.

## 8.1.3 Invoking the Procedure

Once you have made the necessary preparations, you can invoke the cluster configuration procedure to configure your OpenVMS Cluster system. Log in to the system manager account and make sure your default is SYS$MANAGER. Then, invoke the procedure at the DCL command prompt as follows:

```
$ @CLUSTER_CONFIG_LAN
```

or

```
$ @CLUSTER_CONFIG
```

**Caution:** Do not invoke multiple sessions simultaneously. You can run only one cluster configuration session at a time.

Once invoked, both procedures display the following information and menu. (The only difference between CLUSTER_CONFIG_LAN.COM and CLUSTER_CONFIG.COM at this point is the command procedure name that is displayed.) Depending on the menu option you select, the procedure interactively requests configuration information from you. (Predetermine your answers as described in Table 8–3.)

```
                Cluster Configuration Procedure

    Use CLUSTER_CONFIG.COM to set up or change an OpenVMS Cluster configuration.
    To ensure that you have the required privileges, invoke this procedure
    from the system manager's account.

    Enter ? for help at any prompt.

            1. ADD a node to the cluster.
            2. REMOVE a node from the cluster.
            3. CHANGE a cluster member's characteristics.
            4. CREATE a second system disk for JUPITR.
            5. MAKE a directory structure for a new root on a system disk.
            6. DELETE a root from a system disk.

    Enter choice [1]:
   .
   .
   .
```

This chapter contains a number of sample sessions showing how to run the cluster configuration procedures. Although the CLUSTER_CONFIG_LAN.COM and the CLUSTER_CONFIG.COM procedure function the same for both Alpha and VAX systems, the questions and format may appear slightly different according to the type of computer system.

## 8.2 Adding Computers

In most cases, you invoke either CLUSTER_CONFIG_LAN.COM or CLUSTER_CONFIG.COM on an active OpenVMS Cluster computer and select the ADD function to enable a computer as an OpenVMS Cluster member. However, in some circumstances, you may need to perform extra steps to add computers. Use the information in Table 8–4 to determine your actions.

**Table 8–4 Preparing to Add Computers to an OpenVMS Cluster**

| IF... | THEN... |
|---|---|
| You are adding your first satellite node to the OpenVMS Cluster. | Follow these steps:<br>1. Log in to the computer that will be enabled as the cluster boot server.<br>2. Invoke the cluster configuration procedure, and execute the CHANGE function described in Section 8.4 to enable the local computer as a boot server.<br>3. After the CHANGE function completes, execute the ADD function to add satellites to the cluster. |

(continued on next page)

**Table 8–4 (Cont.) Preparing to Add Computers to an OpenVMS Cluster**

| IF... | THEN... |
|---|---|
| The cluster uses DECdtm services. | You must create a transaction log for the computer when you have configured it into your cluster. For step-by-step instructions on how to do this, see the chapter on DECdtm services in the *OpenVMS System Manager's Manual*. |
| You add a CI connected computer that boots from a cluster common system disk. | You must create a new default bootstrap command procedure for the computer before booting it into the cluster. For instructions, refer to your computer-specific installation and operations guide. |
| You are adding computers to a cluster with more than one common system disk. | You must use a different device name for each system disk on which computers are added. For this reason, the cluster configuration procedure supplies as a default device name the logical volume name (for example, DISK$MARS_SYS1) of SYS$SYSDEVICE: on the local system.<br>Using different device names ensures that each computer added has a unique root directory specification, even if the system disks contain roots with the same name—for example, DISK$MARS_SYS1:[SYS10] and DISK$MARS_SYS2:[SYS10]. |
| You add a voting member to the cluster. | You must, after the ADD function completes, reconfigure the cluster according to the instructions in Section 8.6. |

**Caution:** If either the local or the new computer fails before the ADD function completes, you must, after normal conditions are restored, perform the REMOVE option to erase any invalid data and then restart the ADD option. Section 8.3 describes the REMOVE option.

### 8.2.1 Controlling Conversational Bootstrap Operations

When you add a satellite to the cluster using either cluster configuration command procedure, the procedure asks whether you want to allow conversational bootstrap operations for the satellite (default is No).

If you select the default, the NISCS_CONV_BOOT system parameter in the satellite's system parameter file remains set to 0 to disable such operations. The parameter file (ALPHAVMSSYS.PAR for Alpha systems or VAXVMSSYS.PAR for VAX systems) resides in the satellite's root directory on a boot server's system disk (*device*:[SYS*x*.SYSEXE]). You can enable conversational bootstrap operations for a given satellite at any time by setting this parameter to 1.

**Example:**

To enable such operations for an OpenVMS VAX satellite booted from root 10 on device $1$DJA11, you would proceed as follows:

| Step | Action |
|---|---|
| **1** | Log in as system manager on the boot server. |
| **2** | †On VAX systems, invoke the System Generation utility (SYSGEN) and enter the following commands:<br>`$ RUN SYS$SYSTEM:SYSGEN`<br>`SYSGEN> USE $1$DJA11:[SYS10.SYSEXE]VAXVMSSYS.PAR`<br>`SYSGEN> SET NISCS_CONV_BOOT 1`<br>`SYSGEN> WRITE $1$DJA11:[SYS10.SYSEXE]VAXVMSSYS.PAR`<br>`SYSGEN> EXIT`<br>`$` |

†VAX specific

| Step | Action |
|---|---|
| | ‡On an Alpha satellite, enter the same commands, replacing VAXVMSSYS.PAR with ALPHAVMSSYS.PAR. |
| **3** | Modify the satellite's MODPARAMS.DAT file so that NISCS_CONV_BOOT is set to 1. |

‡Alpha specific

### 8.2.2 Common AUTOGEN Parameter Files

When adding a node or a satellite to an OpenVMS Cluster, the cluster configuration command procedure adds one of the following lines in the MODPARAMS.DAT file:

| WHEN the node being added is a... | THEN... |
|---|---|
| Satellite node | The following line is added to the MODPARAMS.DAT file:<br>`AGEN$INCLUDE_PARAMS SYS$MANAGER:AGEN$NEW_SATELLITE_DEFAULTS.DAT` |
| Nonsatellite node | The following line is added to the MODPARAMS.DAT file:<br>`AGEN$INCLUDE_PARAMS SYS$MANAGER:AGEN$NEW_NODE_DEFAULTS.DAT` |

The AGEN$NEW_SATELLITE_DEFAULTS.DAT and AGEN$NEW_NODE_DEFAULTS.DAT files hold AUTOGEN parameter settings that are common to all satellite nodes or nonsatellite nodes in the cluster. Use of these files simplifies system management, because you can maintain common system parameters in either one or both of these files. When adding or changing the common parameters, this eliminates the need to make modifications in the MODPARAMS.DAT files located on every node in the cluster.

Initially, these files contain no parameter settings. You edit the AGEN$NEW_SATELLITE_DEFAULTS.DAT and AGEN$NEW_NODE_DEFAULTS.DAT files, as appropriate, to add, modify, or edit system parameters. For example, you might edit the AGEN$NEW_SATELLITE_DEFAULTS.DAT file to set the MIN_GBLPAGECNT parameter to 5000. AUTOGEN makes the MIN_GBLPAGECNT parameter and all other parameter settings in the AGEN$NEW_SATELLITE_DEFAULTS.DAT file common to all satellite nodes in the cluster.

AUTOGEN uses the parameter settings in the AGEN$NEW_SATELLITE_DEFAULTS.DAT or AGEN$NEW_NODE_DEFAULTS.DAT files the first time it is run, and with every subsequent execution.

### 8.2.3 Examples

Examples 8–1, 8–2, and 8–3 illustrate the use of CLUSTER_CONFIG.COM on JUPITR to add, respectively, a boot server running DECnet for OpenVMS, a boot server running DECnet–Plus, and a satellite node.

### Example 8–1 Sample Interactive CLUSTER_CONFIG.COM Session to Add a Computer as a Boot Server

```
$ @CLUSTER_CONFIG.COM

                Cluster Configuration Procedure

    Use CLUSTER_CONFIG.COM to set up or change an OpenVMS Cluster configuration.
    To ensure that you have the required privileges, invoke this procedure
    from the system manager's account.
    Enter ? for help at any prompt.

            1. ADD a node to the cluster.
            2. REMOVE a node from the cluster.
            3. CHANGE a cluster node's characteristics.
            4. CREATE a second system disk for JUPITR.
            5. MAKE a directory structure for a new root on a system disk.
            6. DELETE a root from a system disk.

    Enter choice [1]: Return

    The ADD function adds a new node to the cluster.

    If the node being added is a voting member, EXPECTED_VOTES in all
    other cluster members' MODPARAMS.DAT must be adjusted. You must then
    reconfigure the cluster, using the procedure described in the
    OpenVMS Cluster Systems manual.

    If the new node is a satellite, the network databases on JUPITR are
    updated. The network databases on all other cluster members must be
    updated.

    For instructions, see the OpenVMS Cluster Systems manual.

What is the node's DECnet node name? SATURN
What is the node's DECnet address? 2.3
Will SATURN be a satellite [Y]? N
Will SATURN be a boot server [Y]? Return

    This procedure will now ask you for the device name of SATURN's system root.
    The default device name (DISK$VAXVMSRL5:) is the logical volume name of
    SYS$SYSDEVICE:.

What is the device name for SATURN's system root [DISK$VAXVMSRL5:]? Return
What is the name of the new system root [SYSA]? Return
Creating directory tree SYSA...
%CREATE-I-CREATED, $1$DJA11:<SYSA> created
%CREATE-I-CREATED, $1$DJA11:<SYSA.SYSEXE> created
   .
   .
   .
System root SYSA created.
Enter a value for SATURN's ALLOCLASS parameter: 1
Does this cluster contain a quorum disk [N]? Y
What is the device name of the quorum disk? $1$DJA12
Updating network database...
Size of page file for SATURN [10000 blocks]? 50000
Size of swap file for SATURN [8000 blocks]? 20000
Will a local (non-HSC) disk on SATURN be used for paging and swapping? N

     If you specify a device other than DISK$VAXVMSRL5: for SATURN's
     page and swap files, this procedure will create PAGEFILE_SATURN.SYS
     and SWAPFILE_SATURN.SYS in the <SYSEXE> directory on the device you
     specify.
What is the device name for the page and swap files [DISK$VAXVMSRL5:]? Return
%SYSGEN-I-CREATED, $1$DJA11:<SYSA.SYSEXE>PAGEFILE.SYS;1 created
%SYSGEN-I-CREATED, $1$DJA11:<SYSA.SYSEXE>SWAPFILE.SYS;1 created
```

(continued on next page)

**Example 8–1 (Cont.) Sample Interactive CLUSTER_CONFIG.COM Session to Add a Computer as a Boot Server**

```
The configuration procedure has completed successfully.
     SATURN has been configured to join the cluster.

     Before booting SATURN, you must create a new default bootstrap
     command procedure for SATURN. See your processor-specific
     installation and operations guide for instructions.

     The first time SATURN boots, NET$CONFIGURE.COM and
     AUTOGEN.COM will run automatically.

     The following parameters have been set for SATURN:

                  VOTES = 1
                  QDSKVOTES = 1

     After SATURN has booted into the cluster, you must increment
     the value for EXPECTED_VOTES in every cluster member's
     MODPARAMS.DAT. You must then reconfigure the cluster, using the
     procedure described in the OpenVMS Cluster Systems manual.
```

**Example 8–2 Sample Interactive CLUSTER_CONFIG.COM Session to Add a Computer Running DECnet–Plus**

```
$ @CLUSTER_CONFIG.COM

                Cluster Configuration Procedure

    Use CLUSTER_CONFIG.COM to set up or change an OpenVMS Cluster configuration.
    To ensure that you have the required privileges, invoke this procedure
    from the system manager's account.

    Enter ? for help at any prompt.

            1. ADD a node to the cluster.
            2. REMOVE a node from the cluster.
            3. CHANGE a cluster node's characteristics.
            4. CREATE a second system disk for JUPITR.
            5. MAKE a directory structure for a new root on a system disk.
            6. DELETE a root from a system disk.

    Enter choice [1]: Return

    The ADD function adds a new node to the cluster.

    If the node being added is a voting member, EXPECTED_VOTES in all
    other cluster members' MODPARAMS.DAT must be adjusted, and the
    cluster must be rebooted.

    If the new node is a satellite, the network databases on JUPITR are
    updated. The network databases on all other cluster members must be
    updated.

    For instructions, see the OpenVMS Cluster Systems manual.

    For additional networking information, please refer to the
    DECnet-Plus Network Management manual.


What is the node's DECnet fullname? OMNI:.DISCOVERY.SATURN
What is the SCS node name for this node? SATURN
Do you want to define a DECnet synonym [Y]? Y
What is the synonym name for this node [SATURN]? SATURN
What is the MOP service client name for this node [SATURN]? VENUS
What is the node's DECnet Phase IV address? 17.129
Will SATURN be a satellite [Y]? N
Will SATURN be a boot server [Y]? Return
```

(continued on next page)

**Example 8–2 (Cont.) Sample Interactive CLUSTER_CONFIG.COM Session to Add a Computer Running DECnet–Plus**

```
    This procedure will now ask you for the device name of SATURN's system root.
    The default device name (DISK$VAXVMSRL5:) is the logical volume name of
    SYS$SYSDEVICE:.

What is the device name for SATURN's system root [DISK$VAXVMSRL5:]? Return
What is the name of the new system root [SYSA]? Return
Creating directory tree SYSA...
%CREATE-I-CREATED, $1$DJA11:<SYSA> created
%CREATE-I-CREATED, $1$DJA11:<SYSA.SYSEXE> created
   .
   .
   .
System root SYSA created.
Enter a value for SATURN's ALLOCLASS parameter: 1
Does this cluster contain a quorum disk [N]? Y
What is the device name of the quorum disk? $1$DJA12
Updating network database...
Size of page file for SATURN [10000 blocks]? 50000
Size of swap file for SATURN [8000 blocks]? 20000
Will a local (non-HSC) disk on SATURN be used for paging and swapping? N

     If you specify a device other than DISK$VAXVMSRL5: for SATURN's
     page and swap files, this procedure will create PAGEFILE_SATURN.SYS
     and SWAPFILE_SATURN.SYS in the <SYSEXE> directory on the device you
     specify.
What is the device name for the page and swap files [DISK$VAXVMSRL5:]? Return
%SYSGEN-I-CREATED, $1$DJA11:<SYSA.SYSEXE>PAGEFILE.SYS;1 created
%SYSGEN-I-CREATED, $1$DJA11:<SYSA.SYSEXE>SWAPFILE.SYS;1 created
The configuration procedure has completed successfully.
     SATURN has been configured to join the cluster.

     Before booting SATURN, you must create a new default bootstrap
     command procedure for SATURN. See your processor-specific
     installation and operations guide for instructions.

     The first time SATURN boots, NETCONFIG.COM and
     AUTOGEN.COM will run automatically.

     The following parameters have been set for SATURN:

                  VOTES = 1
                  QDSKVOTES = 1

     After SATURN has booted into the cluster, you must increment
     the value for EXPECTED_VOTES in every cluster member's
     MODPARAMS.DAT. You must then reconfigure the cluster, using the
     procedure described in the OpenVMS Cluster Systems manual.
```

### Example 8–3 Sample Interactive CLUSTER_CONFIG.COM Session to Add a VAX Satellite with Local Page and Swap Files

```
$ @CLUSTER_CONFIG.COM

                Cluster Configuration Procedure

    Use CLUSTER_CONFIG.COM to set up or change a OpenVMS Cluster configuration.
    To ensure that you have the required privileges, invoke this procedure
    from the system manager's account.

    Enter ? for help at any prompt.

            1. ADD a node to the cluster.
            2. REMOVE a node from the cluster.
            3. CHANGE a cluster node's characteristics.
            4. CREATE a second system disk for JUPITR.
            5. MAKE a directory structure for a new root on a system disk.
            6. DELETE a root from a system disk.

    Enter choice [1]: Return

    The ADD function adds a new node to the cluster.

    If the node being added is a voting member, EXPECTED_VOTES in all
    other cluster members' MODPARAMS.DAT must be adjusted, and the
    cluster must be rebooted.

    If the new node is a satellite, the network databases on JUPITR are
    updated. The network databases on all other cluster members must be
    updated.

    For instructions, see the OpenVMS Cluster Systems manual.

What is the node's DECnet node name? EUROPA
What is the node's DECnet address? 2.21
Will EUROPA be a satellite [Y]? Return
Verifying circuits in network database...

    This procedure will now ask you for the device name of EUROPA's system root.
    The default device name (DISK$VAXVMSRL5:) is the logical volume name of
    SYS$SYSDEVICE:.

What is the device name for EUROPA'S system root [DISK$VAXVMSRL5:]? Return
What is the name of the new system root [SYS10]? Return
Allow conversational bootstraps on EUROPA [NO]? Return
The following workstation windowing options are available:

     1. No workstation software
     2. VWS Workstation Software
     3. DECwindows Workstation Software

Enter choice [1]: 3

Creating directory tree SYS10...
%CREATE-I-CREATED, $1$DJA11:<SYS10> created
%CREATE-I-CREATED, $1$DJA11:<SYS10.SYSEXE> created
   .
   .
   .
System root SYS10 created.
Will EUROPA be a disk server [N]? Return
What is EUROPA's Ethernet hardware address? 08-00-2B-03-51-75
Updating network database...
Size of pagefile for EUROPA [10000 blocks]? 20000
Size of swap file for EUROPA [8000 blocks]? 12000
Will a local disk on EUROPA be used for paging and swapping? YES
Creating temporary page file in order to boot EUROPA for the first time...
%SYSGEN-I-CREATED, $1$DJA11:<SYS10.SYSEXE>PAGEFILE.SYS;1 created

    This procedure will now wait until EUROPA joins the cluster.

    Once EUROPA joins the cluster, this procedure will ask you
    to specify a local disk on EUROPA for paging and swapping.
```

(continued on next page)

**Example 8–3 (Cont.) Sample Interactive CLUSTER_CONFIG.COM Session to Add a VAX Satellite with Local Page and Swap Files**

```
    Please boot EUROPA now.

Waiting for EUROPA to boot...
   .
   .
   .
(User enters boot command at satellite's console-mode prompt (>>>).)
   .
   .
   .
The local disks on EUROPA are:

Device             Device       Error     Volume    Free    Trans  Mnt
Name               Status       Count     Label     Blocks  Count  Cnt
EUROPA$DUA0:       Online           0
EUROPA$DUA1:       Online           0

Which disk can be used for paging and swapping? EUROPA$DUA0:
May this procedure INITIALIZE EUROPA$DUA0: [YES]? NO
Mounting EUROPA$DUA0:...
PAGEFILE.SYS already exists on EUROPA$DUA0:

****************************************

Directory EUROPA$DUA0:[SYS0.SYSEXE]

PAGEFILE.SYS;1      23600/23600

Total of 1 file, 23600/23600 blocks.

****************************************

What is the file specification for the page file on
EUROPA$DUA0: [ <SYS0.SYSEXE>PAGEFILE.SYS ]? Return
%CREATE-I-EXISTS, EUROPA$DUA0:<SYS0.SYSEXE> already exists
This procedure will use the existing pagefile,
EUROPA$DUA0:<SYS0.SYSEXE>PAGEFILE.SYS;.

SWAPFILE.SYS already exists on EUROPA$DUA0:

****************************************

Directory EUROPA$DUA0:[SYS0.SYSEXE]

SWAPFILE.SYS;1      12000/12000

Total of 1 file, 12000/12000 blocks.

****************************************

What is the file specification for the swap file on
EUROPA$DUA0: [ <SYS0.SYSEXE>SWAPFILE.SYS ]? Return
This procedure will use the existing swapfile,
EUROPA$DUA0:<SYS0.SYSEXE>SWAPFILE.SYS;.

     AUTOGEN will now reconfigure and reboot EUROPA automatically.
     These operations will complete in a few minutes, and a
     completion message will be displayed at your terminal.

The configuration procedure has completed successfully.
```

## 8.2.4 Adding a Quorum Disk

To enable a quorum disk on a node or nodes, use the cluster configuration procedure as described in Table 8–5.

**Table 8–5 Preparing to Add a Quorum Disk Watcher**

| IF... | THEN... |
|---|---|
| Other cluster nodes are already enabled as quorum disk watchers. | Perform the following steps:<br>1. Log in to the computer that is to be enabled as the quorum disk watcher and run CLUSTER_CONFIG_LAN.COM or CLUSTER_CONFIG.COM.<br>2. Execute the CHANGE function and select menu item 7 to enable a quorum disk. (See Section 8.4.)<br>3. Update the current system parameters and reboot the node. (See Section 8.6.1.) |
| The cluster does not contain any quorum disk watchers. | Perform the following steps:<br>1. Perform the preceding steps 1 and 2 for each node to be enabled as a quorum disk watcher.<br>2. Reconfigure the cluster according to the instructions in Section 8.6. |

## 8.3 Removing Computers

To disable a computer as an OpenVMS Cluster member:

1. Determine whether removing a member will cause you to lose quorum. Use the SHOW CLUSTER command to display the CL_QUORUM and CL_VOTES values.

| IF removing members... | THEN... |
|---|---|
| Will cause you to lose quorum | Perform the steps in the following list:<br>**Caution:** Do not perform these steps until you are ready to reboot the entire OpenVMS Cluster system. Because you are reducing quorum for the cluster, the votes cast by the node being removed could cause a cluster partition to be formed.<br>• Reset the EXPECTED_VOTES parameter in the AUTOGEN parameter files and current system parameter files (see Section 8.6.1).<br>• Shut down the cluster (see Section 8.6.2), and reboot without the node that is being removed.<br>**Note:** Be sure that you do not specify an automatic reboot on that node. |
| Will not cause you to lose quorum | Proceed as follows:<br>• Perform an orderly shutdown on the node being removed by invoking the SYS$SYSTEM:SHUTDOWN.COM command procedure (described in Section 8.6.3).<br>• If the node was a voting member, use the DCL command SET CLUSTER/EXPECTED_VOTES to reduce the value of quorum. |

**Reference:** Refer also to Section 10.12 for information about adjusting expected votes.

2. Invoke CLUSTER_CONFIG_LAN.COM or CLUSTER_CONFIG.COM on an active OpenVMS Cluster computer and select the REMOVE option.
3. Use the information in Table 8–6 to determine whether additional actions are required.

**Table 8–6 Preparing to Remove Computers from an OpenVMS Cluster**

| IF... | THEN... |
|---|---|
| You are removing a voting member. | You must, after the REMOVE function completes, reconfigure the cluster according to the instructions in Section 8.6. |
| The page and swap files for the computer being removed do not reside on the same disk as the computer's root directory tree. | The REMOVE function does not delete these files. It displays a message warning that the files will not be deleted, as in Example 8–4. If you want to delete the files, you must do so after the REMOVE function completes. |
| You are removing a computer from a cluster that uses DECdtm services. | Make sure that you have followed the step-by-step instructions in the chapter on DECdtm services in the *OpenVMS System Manager's Manual*. These instructions describe how to remove a computer safely from the cluster, thereby preserving the integrity of your data. |

**Note:** When the REMOVE function deletes the computer's entire root directory tree, it generates OpenVMS RMS informational messages while deleting the directory files. You can ignore these messages.

### 8.3.1 Example

Example 8–4 illustrates the use of CLUSTER_CONFIG.COM on JUPITR to remove satellite EUROPA from the cluster.

**Example 8–4 Sample Interactive CLUSTER_CONFIG.COM Session to Remove a Satellite with Local Page and Swap Files**

```
$ @CLUSTER_CONFIG.COM

                Cluster Configuration Procedure

    Use CLUSTER_CONFIG.COM to set up or change an OpenVMS Cluster configuration.
    To ensure that you have the required privileges, invoke this procedure
    from the system manager's account.

    Enter ? for help at any prompt.

            1. ADD a node to the cluster.
            2. REMOVE a node from the cluster.
            3. CHANGE a cluster node's characteristics.
            4. CREATE a second system disk for JUPITR.
            5. MAKE a directory structure for a new root on a system disk.
            6. DELETE a root from a system disk.

    Enter choice [1]: 2
```

(continued on next page)

**Example 8–4 (Cont.) Sample Interactive CLUSTER_CONFIG.COM Session to Remove a Satellite with Local Page and Swap Files**

```
    The REMOVE function disables a node as a cluster member.

        o It deletes the node's root directory tree.

        o It removes the node's network information
          from the network database.

    If the node being removed is a voting member, you must adjust
    EXPECTED_VOTES in each remaining cluster member's MODPARAMS.DAT.
    You must then reconfigure the cluster, using the procedure described
    in the OpenVMS Cluster Systems manual.

What is the node's DECnet node name? EUROPA
Verifying network database...
Verifying that SYS10 is EUROPA's root...

    WARNING - EUROPA's page and swap files will not be deleted.
              They do not reside on $1$DJA11:.

Deleting directory tree SYS10...
%DELETE-I-FILDEL, $1$DJA11:<SYS10>SYSCBI.DIR;1 deleted (1 block)
%DELETE-I-FILDEL, $1$DJA11:<SYS10>SYSERR.DIR;1 deleted (1 block)
   .
   .
   .
System root SYS10 deleted.
Updating network database...
The configuration procedure has completed successfully.
```

## 8.3.2 Removing a Quorum Disk

To disable a quorum disk on a node or nodes, use the cluster configuration command procedure as described in Table 8–7.

**Table 8–7 Preparing to Remove a Quorum Disk Watcher**

| IF... | THEN... |
|---|---|
| Other cluster nodes will still be enabled as quorum disk watchers. | Perform the following steps:<br>1. Log in to the computer that is to be disabled as the quorum disk watcher and run CLUSTER_CONFIG_LAN.COM or CLUSTER_CONFIG.COM.<br>2. Execute the CHANGE function and select menu item 7 to disable a quorum disk (see Section 8.4).<br>3. Reboot the node (see Section 8.6.7). |
| All quorum disk watchers will be disabled. | Perform the following steps:<br>1. Perform the preceding steps 1 and 2 for all computers with the quorum disk enabled.<br>2. Reconfigure the cluster according to the instructions in Section 8.6. |

## 8.4 Changing Computer Characteristics

As your processing needs change, you may want to add satellites to an existing OpenVMS Cluster, or you may want to change an OpenVMS Cluster that is based on one interconnect (such as the CI or DSSI interconnect, or HSC subsystem) to include several interconnects.

Table 8–8 describes the operations you can accomplish when you select the CHANGE option from the main menu of the cluster configuration command procedure.

**Note:** All operations except changing a satellite's LAN (Ethernet or FDDI) hardware address must be executed on the computer whose characteristics you want to change.

**Table 8–8 CHANGE Options of the Cluster Configuration Procedure**

| Option | Operation Performed |
|---|---|
| Enable the local computer as a disk server | Loads the MSCP server by setting, in MODPARAMS.DAT, the value of the MSCP_LOAD parameter to 1 and the MSCP_SERVE_ALL parameter to 1 or 2. |
| Disable the local computer as a disk server | Sets MSCP_LOAD to 0. |
| Enable the local computer as a boot server | If you are setting up an OpenVMS Cluster that includes satellites, you must perform this operation once before you attempt to add satellites to the cluster. You thereby enable MOP service for the LAN adapter circuit that the computer uses to service operating system load requests from satellites. When you enable the computer as a boot server, it automatically becomes a disk server (if it is not one already) because it must serve its system disk to satellites. |
| Disable the local computer as a boot server | Disables DECnet MOP service for the computer's adapter circuit. |
| Enable the LAN for cluster communications on the local computer | Loads the port driver PEDRIVER by setting the value of the NISCS_LOAD_PEA0 parameter to 1 in MODPARAMS.DAT. Creates the cluster security database file, SYS$SYSTEM:[SYSEXE]CLUSTER_AUTHORIZE.DAT, on the local computer's system disk.<br>**Caution:** The VAXCLUSTER system parameter must be set to 2 if the NISCS_LOAD_PEA0 parameter is set to 1. This ensures coordinated access to shared resources in the cluster and prevents accidental data corruption. |
| Disable the LAN for cluster communications on the local computer | Sets NISCS_LOAD_PEA0 to 0. |
| Enable a quorum disk on the local computer | In MODPARAMS.DAT, sets the DISK_QUORUM system parameter to a device name; sets the value of QDSKVOTES to 1 (default value). |
| Disable a quorum disk on the local computer | In MODPARAMS.DAT, sets a blank value for the DISK_QUORUM system parameter; sets the value of QDSKVOTES to 1. |
| Change a satellite's LAN hardware address | Changes a satellite's hardware address if its LAN device needs replacement. Both the permanent and volatile network databases and NETNODE_UPDATE.COM are updated on the local computer.<br>**Rule:** *You must perform this operation on each computer enabled as a boot server for the satellite.* |
| Enable the local computer as a tape server | Loads the TMSCP server by setting, in MODPARAMS.DAT, the value of the TMSCP_LOAD parameter to 1 and the TMSCP_SERVE_ALL parameter to 1 or 2. |

(continued on next page)

**Table 8–8 (Cont.) CHANGE Options of the Cluster Configuration Procedure**

| Option | Operation Performed |
|---|---|
| Disable the local computer as a tape server | Sets TMSCP_LOAD to zero. |
| Change the local computer's node allocation class value | Sets a value for the computer's ALLOCLASS parameter in MODPARAMS.DAT. |
| Change the local computer's tape allocation class value | Sets a value from 1 to 255 for the computer's TAPE_ALLOCLASS parameter in MODPARAMS.DAT. The default value is zero. You must specify a nonzero tape allocation class parameter if this node is locally connected to a dual-ported tape, or if it will be serving any multiple-host tapes (for example, TF*nn* or HSC connected tapes) to other cluster members. Satellites usually have TAPE_ALLOCLASS set to zero. |
| Change the local computer's port allocation class value | Sets a value for the computer's ALLOCLASS parameter in MODPARAMS.DAT for all devices attached to it. |
| Enable MEMORY CHANNEL | Sets MC_SERVICES_P2 to 1 to load the PMDRIVER (PMA0) cluster driver. This system parameter enables MEMORY CHANNEL on the local computer for node-to-node cluster communications. |
| Disable MEMORY CHANNEL | Sets MC_SERVICES_P2 to 0 so that the PMDRIVER (PMA0) cluster driver is not loaded. The setting of 0 disables MEMORY CHANNEL on the local computer as the node-to-node cluster communications interconnect. |

### 8.4.1 Preparation

You usually need to perform a number of steps before using the cluster configuration command procedure to change the configuration of your existing cluster.

Table 8–9 suggests several typical configuration changes and describes the procedures required to make them.

**Table 8–9 Tasks Involved in Changing OpenVMS Cluster Configurations**

| Task | Procedure |
|---|---|
| Add satellite nodes | Perform these operations on the computer that will be enabled as a cluster boot server:<br>1. Execute the CHANGE function to enable the first installed computer as a boot server (see Example 8–7).<br>2. Execute the ADD function to add the satellite (as described in Section 8.2).<br>3. Reconfigure the cluster according to the postconfiguration instructions in Section 8.6. |
| Change an existing CI or DSSI cluster to include satellite nodes | To enable cluster communications over the LAN (Ethernet or FDDI) on all computers, and to enable one or more computers as boot servers, proceed as follows: |

(continued on next page)

**Table 8–9 (Cont.) Tasks Involved in Changing OpenVMS Cluster Configurations**

| Task | Procedure |
|---|---|
| | 1. Log in as system manager on each computer, invoke either CLUSTER_CONFIG_LAN.COM or CLUSTER_CONFIG.COM, and execute the CHANGE function to enable LAN communications.<br><br>**Rule:** *You must perform this operation on all computers.*<br><br>**Note:** You must establish a cluster group number and password on all system disks in the OpenVMS Cluster before you can successfully add a satellite node using the CHANGE function of the cluster configuration procedure.<br><br>2. Execute the CHANGE function to enable one or more computers as boot servers.<br><br>3. Reconfigure the cluster according to the postconfiguration instructions in Section 8.6. |
| Change an existing LAN-based cluster to include CI and DSSI interconnects | Before performing the operations described here, be sure that the computers and HSC subsystems or RF disks you intend to include in your new configuration are correctly installed and checked for proper operation.<br><br>The method you use to include CI and DSSI interconnects with an existing LAN-based cluster configuration depends on whether your current boot server is capable of being configured as a CI or DSSI computer.<br><br>(continued on next page) |

**Table 8–9 (Cont.) Tasks Involved in Changing OpenVMS Cluster Configurations**

| Task | Procedure |
|---|---|
| | **Note:** The following procedures assume that the system disk containing satellite roots will reside on an HSC disk (for CI configurations) or an RF disk (for DSSI configurations).<br>• If the boot server can be configured as a CI or DSSI computer, proceed as follows:<br>1. Log in as system manager on the boot server and perform an image backup operation to back up the current system disk to a disk on an HSC subsystem or RF storage device. (For more information about backup operations, refer to the *OpenVMS System Management Utilities Reference Manual*.)<br>2. Modify the computer's default bootstrap command procedure to boot the computer from the HSC or RF disk, according to the instructions in the appropriate system-specific installation and operations guide.<br>3. Shut down the cluster. Shut down the satellites first, and then shut down the boot server.<br>4. Boot the boot server from the newly created system disk on the HSC or RF storage subsystem.<br>5. Reboot the satellites.<br>• If your current boot server cannot be configured as a CI or a DSSI computer, proceed as follows:<br>1. Shut down the old local area cluster. Shut down the satellites first, and then shut down the boot server.<br>2. Install the OpenVMS operating system on the new CI computer's HSC system disk or on the new DSSI computer's RF disk, as appropriate. When the installation procedure asks whether you want to enable the LAN for cluster communications, answer YES.<br>3. When the installation completes, log in as system manager, and configure and start the DECnet for OpenVMS network as described in Chapter 4.<br>4. Execute the CHANGE function of either CLUSTER_CONFIG_LAN.COM or CLUSTER_CONFIG.COM to enable the computer as a boot server.<br>5. Log in as system manager on the newly added computer and execute the ADD function of either CLUSTER_CONFIG_LAN.COM or CLUSTER_CONFIG.COM to add the former LAN cluster members (including the former boot server) as satellites.<br>• Reconfigure the cluster according to the postconfiguration instructions in Section 8.6. |

(continued on next page)

**Table 8–9 (Cont.) Tasks Involved in Changing OpenVMS Cluster Configurations**

| Task | Procedure |
|---|---|
| Convert a standalone computer to an OpenVMS Cluster computer | Execute either CLUSTER_CONFIG_LAN.COM or CLUSTER_CONFIG.COM on a standalone computer to perform either of the following operations:<br>• Add the standalone computer with its own system disk to an existing cluster.<br>• Set up the standalone computer to form a new cluster if the computer was not set up as a cluster computer during installation of the operating system.<br>• Reconfigure the cluster according to the postconfiguration instructions in Section 8.6.<br>See Example 8–11, which illustrates the use of CLUSTER_CONFIG.COM on standalone computer PLUTO to convert PLUTO to a cluster boot server.<br>If your cluster uses DECdtm services, you must create a transaction log for the computer when you have configured it into your cluster. For step-by-step instructions on how to do this, see the chapter on DECdtm services in the *OpenVMS System Manager's Manual*. |
| Enable or disable disk-serving or tape-serving functions | After invoking either CLUSTER_CONFIG_LAN.COM or CLUSTER_CONFIG.COM to enable or disable the disk or tape serving functions, run AUTOGEN with the REBOOT option to reboot the local computer (see Section 8.6.1). |

**Note:** When the cluster configuration command procedure sets or changes values in MODPARAMS.DAT, the new values are always appended at the end of the file so that they override earlier values. You may want to edit the file occasionally and delete lines that specify earlier values.

### 8.4.2 Examples

Examples 8–5 through 8–11 show the use of CLUSTER_CONFIG.COM to perform the following operations:

- Enable a computer as a disk server (Example 8–5).
- Change a computer's ALLOCLASS value (Example 8–6).
- Enable a computer as a boot server (Example 8–7).
- Specify a new hardware address for a satellite node that boots from a common system disk (Example 8–8).
- Enable a computer as a tape server (Example 8–9).
- Change a computer's TAPE_ALLOCLASS value (Example 8–10).
- Convert a standalone computer to a cluster boot server (Example 8–11).

**Example 8–5 Sample Interactive CLUSTER_CONFIG.COM Session to Enable the Local Computer as a Disk Server**

```
$ @CLUSTER_CONFIG.COM

                Cluster Configuration Procedure

    Use CLUSTER_CONFIG.COM to set up or change an OpenVMS Cluster configuration.
    To ensure that you have the required privileges, invoke this procedure
    from the system manager's account.

    Enter ? for help at any prompt.
```

(continued on next page)

**Example 8–5 (Cont.) Sample Interactive CLUSTER_CONFIG.COM Session to Enable the Local Computer as a Disk Server**

```
            1. ADD a node to the cluster.
            2. REMOVE a node from the cluster.
            3. CHANGE a cluster node's characteristics.
            4. CREATE a second system disk for URANUS.
            5. MAKE a directory structure for a new root on a system disk.
            6. DELETE a root from a system disk.

    Enter choice [1]: 3

    CHANGE Menu

       1. Enable URANUS as a disk server.
       2. Disable URANUS as a disk server.
       3. Enable URANUS as a boot server.
       4. Disable URANUS as a boot server.
       5. Enable the LAN for cluster communications on URANUS.
       6. Disable the LAN for cluster communications on URANUS.
       7. Enable a quorum disk on URANUS.
       8. Disable a quorum disk on URANUS.
       9. Change URANUS's satellite's Ethernet or FDDI hardware address.
      10. Enable URANUS as a tape server.
      11. Disable URANUS as a tape server.
      12. Change URANUS's ALLOCLASS value.
      13. Change URANUS's TAPE_ALLOCLASS value.
      14. Change URANUS's shared SCSI port allocation class value.
      15. Enable Memory Channel for cluster communications on Uranus.
      16. Disable Memory Channel for cluster communications on Uranus.

    Enter choice [1]: Return

Will URANUS serve HSC disks [Y]? Return
Enter a value for URANUS's ALLOCLASS parameter: 2
The configuration procedure has completed successfully.

    URANUS has been enabled as a disk server. MSCP_LOAD has been
    set to 1 in MODPARAMS.DAT. Please run AUTOGEN to reboot URANUS:

           $ @SYS$UPDATE:AUTOGEN GETDATA REBOOT

    If you have changed URANUS's ALLOCLASS value, you must reconfigure the
    cluster, using the procedure described in the OpenVMS Cluster Systems manual.
```

**Example 8–6 Sample Interactive CLUSTER_CONFIG.COM Session to Change the Local Computer's ALLOCLASS Value**

```
$ @CLUSTER_CONFIG.COM

                Cluster Configuration Procedure

    Use CLUSTER_CONFIG.COM to set up or change an OpenVMS Cluster configuration.
    To ensure that you have the required privileges, invoke this procedure
    from the system manager's account.

    Enter ? for help at any prompt.

            1. ADD a node to the cluster.
            2. REMOVE a node from the cluster.
            3. CHANGE a cluster node's characteristics.
            4. CREATE a second system disk for URANUS.
            5. MAKE a directory structure for a new root on a system disk.
            6. DELETE a root from a system disk.

    Enter choice [1]: 3
```

(continued on next page)

### Example 8–6 (Cont.) Sample Interactive CLUSTER_CONFIG.COM Session to Change the Local Computer's ALLOCLASS Value

```
    CHANGE Menu

       1. Enable URANUS as a disk server.
       2. Disable URANUS as a disk server.
       3. Enable URANUS as a boot server.
       4. Disable URANUS as a boot server.
       5. Enable the LAN for cluster communications on URANUS.
       6. Disable the LAN for cluster communications on URANUS.
       7. Enable a quorum disk on URANUS.
       8. Disable a quorum disk on URANUS.
       9. Change URANUS's satellite's Ethernet or FDDI hardware address.
      10. Enable URANUS as a tape server.
      11. Disable URANUS as a tape server.
      12. Change URANUS's ALLOCLASS value.
      13. Change URANUS's TAPE_ALLOCLASS value.
      14. Change URANUS's shared SCSI port allocation class value.
      15. Enable Memory Channel for cluster communications on Uranus.
      16. Disable Memory Channel for cluster communications on Uranus.

    Enter choice [1]: 12

Enter a value for URANUS's ALLOCLASS parameter [2]: 1
The configuration procedure has completed successfully

    If you have changed URANUS's ALLOCLASS value, you must reconfigure the
    cluster, using the procedure described in the OpenVMS Cluster Systems manual.
```

### Example 8–7 Sample Interactive CLUSTER_CONFIG.COM Session to Enable the Local Computer as a Boot Server

```
$ @CLUSTER_CONFIG.COM

                Cluster Configuration Procedure

    Use CLUSTER_CONFIG.COM to set up or change an OpenVMS Cluster configuration.
    To ensure that you have the required privileges, invoke this procedure
    from the system manager's account.

    Enter ? for help at any prompt.

            1. ADD a node to the cluster.
            2. REMOVE a node from the cluster.
            3. CHANGE a cluster node's characteristics.
            4. CREATE a second system disk for URANUS.
            5. MAKE a directory structure for a new root on a system disk.
            6. DELETE a root from a system disk.

    Enter choice [1]: 3
```

(continued on next page)

**Example 8–7 (Cont.) Sample Interactive CLUSTER_CONFIG.COM Session to Enable the Local Computer as a Boot Server**

```
    CHANGE Menu

       1. Enable URANUS as a disk server.
       2. Disable URANUS as a disk server.
       3. Enable URANUS as a boot server.
       4. Disable URANUS as a boot server.
       5. Enable the LAN for cluster communications on URANUS.
       6. Disable the LAN for cluster communications on URANUS.
       7. Enable a quorum disk on URANUS.
       8. Disable a quorum disk on URANUS.
       9. Change URANUS's satellite's Ethernet or FDDI hardware address.
      10. Enable URANUS as a tape server.
      11. Disable URANUS as a tape server.
      12. Change URANUS's ALLOCLASS value.
      13. Change URANUS's TAPE_ALLOCLASS value.
      14. Change URANUS's shared SCSI port allocation class value.
      15. Enable Memory Channel for cluster communications on Uranus.
      16. Disable Memory Channel for cluster communications on Uranus.


    Enter choice [1]: 3

Verifying circuits in network database...
Updating permanent network database...

    In order to enable or disable DECnet MOP service in the volatile
    network database, DECnet traffic must be interrupted temporarily.

    Do you want to proceed [Y]? Return

Enter a value for URANUS's ALLOCLASS parameter [1]: Return
The configuration procedure has completed successfully.

    URANUS has been enabled as a boot server. Disk serving and
    Ethernet capabilities are enabled automatically. If URANUS was
    not previously set up as a disk server, please run AUTOGEN to
    reboot URANUS:

           $ @SYS$UPDATE:AUTOGEN GETDATA REBOOT

    If you have changed URANUS's ALLOCLASS value, you must reconfigure the
    cluster, using the procedure described in the OpenVMS Cluster Systems
manual.
```

**Example 8–8 Sample Interactive CLUSTER_CONFIG.COM Session to Change a Satellite's Hardware Address**

```
$ @CLUSTER_CONFIG.COM

                Cluster Configuration Procedure

    Use CLUSTER_CONFIG.COM to set up or change an OpenVMS Cluster configuration.
    To ensure that you have the required privileges, invoke this procedure
    from the system manager's account.

    Enter ? for help at any prompt.

           1. ADD a node to the cluster.
           2. REMOVE a node from the cluster.
           3. CHANGE a cluster node's characteristics.
           4. CREATE a second system disk for URANUS.
           5. MAKE a directory structure for a new root on a system disk.
           6. DELETE a root from a system disk.

    Enter choice [1]: 3
```

(continued on next page)

**Example 8–8 (Cont.) Sample Interactive CLUSTER_CONFIG.COM Session to Change a Satellite's Hardware Address**

```
    CHANGE Menu

       1. Enable URANUS as a disk server.
       2. Disable URANUS as a disk server.
       3. Enable URANUS as a boot server.
       4. Disable URANUS as a boot server.
       5. Enable the LAN for cluster communications on URANUS.
       6. Disable the LAN for cluster communications on URANUS.
       7. Enable a quorum disk on URANUS.
       8. Disable a quorum disk on URANUS.
       9. Change URANUS's satellite's Ethernet or FDDI hardware address.
      10. Enable URANUS as a tape server.
      11. Disable URANUS as a tape server.
      12. Change URANUS's ALLOCLASS value.
      13. Change URANUS's TAPE_ALLOCLASS value.
      14. Change URANUS's shared SCSI port allocation class value.
      15. Enable Memory Channel for cluster communications on Uranus.
      16. Disable Memory Channel for cluster communications on Uranus.

    Enter choice [1]: 9

What is the node's DECnet node name? ARIEL
What is the new hardware address [XX-XX-XX-XX-XX-XX]? 08-00-3B-05-37-78
Updating network database...
The configuration procedure has completed successfully.
```

**Example 8–9 Sample Interactive CLUSTER_CONFIG.COM Session to Enable the Local Computer as a Tape Server**

```
$ @CLUSTER_CONFIG.COM

                Cluster Configuration Procedure

    Use CLUSTER_CONFIG.COM to set up or change an OpenVMS Cluster configuration.
    To ensure that you have the required privileges, invoke this procedure
    from the system manager's account.

    Enter ? for help at any prompt.

            1. ADD a node to the cluster.
            2. REMOVE a node from the cluster.
            3. CHANGE a cluster node's characteristics.
            4. CREATE a second system disk for URANUS.
            5. MAKE a directory structure for a new root on a system disk.
            6. DELETE a root from a system disk.

    Enter choice [1]: 3
```

(continued on next page)

**Example 8–9 (Cont.) Sample Interactive CLUSTER_CONFIG.COM Session to Enable the Local Computer as a Tape Server**

```
    CHANGE Menu

       1. Enable URANUS as a disk server.
       2. Disable URANUS as a disk server.
       3. Enable URANUS as a boot server.
       4. Disable URANUS as a boot server.
       5. Enable the LAN for cluster communications on URANUS.
       6. Disable the LAN for cluster communications on URANUS.
       7. Enable a quorum disk on URANUS.
       8. Disable a quorum disk on URANUS.
       9. Change URANUS's satellite's Ethernet or FDDI hardware address.
      10. Enable URANUS as a tape server.
      11. Disable URANUS as a tape server.
      12. Change URANUS's ALLOCLASS value.
      13. Change URANUS's TAPE_ALLOCLASS value.
      14. Change URANUS's shared SCSI port allocation class value.
      15. Enable Memory Channel for cluster communications on Uranus.
      16. Disable Memory Channel for cluster communications on Uranus.

    Enter choice [1]: 10

Enter a value for URANUS's TAPE_ALLOCLASS parameter [1]: Return

    URANUS has been enabled as a tape server. TMSCP_LOAD has been
    set to 1 in MODPARAMS.DAT. Please run AUTOGEN to reboot URANUS:

           $ @SYS$UPDATE:AUTOGEN GETDATA REBOOT

    If you have changed URANUS's TAPE_ALLOCLASS value, you must reconfigure
    the cluster, using the procedure described in the OpenVMS Cluster Systems
    manual.
```

### Example 8–10 Sample Interactive CLUSTER_CONFIG.COM Session to Change the Local Computer's TAPE_ALLOCLASS Value

```
$ @CLUSTER_CONFIG.COM

                Cluster Configuration Procedure

    Use CLUSTER_CONFIG.COM to set up or change an OpenVMS Cluster configuration.
    To ensure that you have the required privileges, invoke this procedure
    from the system manager's account.

    Enter ? for help at any prompt.

            1. ADD a node to the cluster.
            2. REMOVE a node from the cluster.
            3. CHANGE a cluster node's characteristics.
            4. CREATE a second system disk for URANUS.
            5. MAKE a directory structure for a new root on a system disk.
            6. DELETE a root from a system disk.

    Enter choice [1]: 3

    CHANGE Menu

       1. Enable URANUS as a disk server.
       2. Disable URANUS as a disk server.
       3. Enable URANUS as a boot server.
       4. Disable URANUS as a boot server.
       5. Enable the LAN for cluster communications on URANUS.
       6. Disable the LAN for cluster communications on URANUS.
       7. Enable a quorum disk on URANUS.
       8. Disable a quorum disk on URANUS.
       9. Change URANUS's satellite's Ethernet or FDDI hardware address.
      10. Enable URANUS as a tape server.
      11. Disable URANUS as a tape server.
      12. Change URANUS's ALLOCLASS value.
      13. Change URANUS's TAPE_ALLOCLASS value.
      14. Change URANUS's shared SCSI port allocation class value.
      15. Enable Memory Channel for cluster communications on Uranus.
      16. Disable Memory Channel for cluster communications on Uranus.

    Enter choice [1]: 13

Enter a value for URANUS's TAPE_ALLOCLASS parameter [1]: 2

    If you have changed URANUS's TAPE_ALLOCLASS value, you must reconfigure
    the cluster, using the procedure described in the OpenVMS Cluster Systems
    manual.)
```

**Example 8–11 Sample Interactive CLUSTER_CONFIG.COM Session to Convert a Standalone Computer to a Cluster Boot Server**

```
$ @CLUSTER_CONFIG.COM

                Cluster Configuration Procedure

    Use CLUSTER_CONFIG.COM to set up or change an OpenVMS Cluster configuration.
    To ensure that you have the required privileges, invoke this procedure
    from the system manager's account.

    Enter ? for help at any prompt.

            1. ADD a node to the cluster.
            2. REMOVE a node from the cluster.
            3. CHANGE a cluster node's characteristics.
            4. CREATE a second system disk for URANUS.
            5. MAKE a directory structure for a new root on a system disk.
            6. DELETE a root from a system disk.

    Enter choice [1]: 3

    CHANGE Menu

       1. Enable URANUS as a disk server.
       2. Disable URANUS as a disk server.
       3. Enable URANUS as a boot server.
       4. Disable URANUS as a boot server.
       5. Enable the LAN for cluster communications on URANUS.
       6. Disable the LAN for cluster communications on URANUS.
       7. Enable a quorum disk on URANUS.
       8. Disable a quorum disk on URANUS.
       9. Change URANUS's satellite's Ethernet or FDDI hardware address.
      10. Enable URANUS as a tape server.
      11. Disable URANUS as a tape server.
      12. Change URANUS's ALLOCLASS value.
      13. Change URANUS's TAPE_ALLOCLASS value.
      14. Change URANUS's shared SCSI port allocation class value.
      15. Enable Memory Channel for cluster communications on Uranus.
      16. Disable Memory Channel for cluster communications on Uranus.


    Enter choice [1]: 3

    This procedure sets up this standalone node to join an existing
    cluster or to form a new cluster.

What is the node's DECnet node name? PLUTO
What is the node's DECnet address? 2.5
Will the Ethernet be used for cluster communications (Y/N)? Y
Enter this cluster's group number: 3378
Enter this cluster's password:
Re-enter this cluster's password for verification:
Will PLUTO be a boot server [Y]? Return
Verifying circuits in network database...
Enter a value for PLUTO's ALLOCLASS parameter: 1
Does this cluster contain a quorum disk [N]? Return

    AUTOGEN computes the SYSGEN parameters for your configuration
    and then reboots the system with the new parameters.
```

## 8.5 Creating a Duplicate System Disk

As you continue to add Alpha computers running on an Alpha common system disk or VAX computers running on a VAX common system disk, you eventually reach the disk's storage or I/O capacity. In that case, you want to add one or more common system disks to handle the increased load.

**Reminder:** Remember that a system disk cannot be shared between VAX and Alpha computers. An Alpha system cannot be created from a VAX system disk, and a VAX system cannot be created from an Alpha system disk.

### 8.5.1 Preparation

You can use either CLUSTER_CONFIG_LAN.COM or CLUSTER_CONFIG.COM to set up additional system disks. *After* you have coordinated cluster common files as described in Chapter 5, proceed as follows:

1. Locate an appropriate scratch disk for use as an additional system disk.
2. Log in as system manager.
3. Invoke either CLUSTER_CONFIG_LAN.COM or CLUSTER_CONFIG.COM and select the CREATE option.

### 8.5.2 Example

As shown in Example 8–12, the cluster configuration command procedure:

1. Prompts for the device names of the current and new system disks.
2. Backs up the current system disk to the new one.
3. Deletes all directory roots (except SYS0) from the new disk.
4. Mounts the new disk clusterwide.

**Note:** OpenVMS RMS error messages are displayed while the procedure deletes directory files. You can ignore these messages.

**Example 8–12 Sample Interactive CLUSTER_CONFIG.COM CREATE Session**

```
$ @CLUSTER_CONFIG.COM

                Cluster Configuration Procedure

    Use CLUSTER_CONFIG.COM to set up or change an OpenVMS Cluster configuration.
    To ensure that you have the required privileges, invoke this procedure
    from the system manager's account.

    Enter ? for help at any prompt.

            1. ADD a node to the cluster.
            2. REMOVE a node from the cluster.
            3. CHANGE a cluster node's characteristics.
            4. CREATE a second system disk for JUPITR.
            5. MAKE a directory structure for a new root on a system disk.
            6. DELETE a root from a system disk.

    Enter choice [1]: 4

    The CREATE function generates a duplicate system disk.

            o It backs up the current system disk to the new system disk.

            o It then removes from the new system disk all system roots.

    WARNING - Do not proceed unless you have defined appropriate
              logical names for cluster common files in your
              site-specific startup procedures. For instructions,
              see the OpenVMS Cluster Systems manual.

              Do you want to continue [N]? YES

    This procedure will now ask you for the device name of JUPITR's system root.
    The default device name (DISK$VAXVMSRL5:) is the logical volume name of
    SYS$SYSDEVICE:.

What is the device name of the current system disk [DISK$VAXVMSRL5:]? Return
```

(continued on next page)

**Example 8–12 (Cont.) Sample Interactive CLUSTER_CONFIG.COM CREATE Session**

```
What is the device name for the new system disk? $1$DJA16:
%DCL-I-ALLOC, _$1$DJA16: allocated
%MOUNT-I-MOUNTED, SCRATCH mounted on _$1$DJA16:
What is the unique label for the new system disk [JUPITR_SYS2]? Return

Backing up the current system disk to the new system disk...

Deleting all system roots...

              Deleting directory tree SYS1...

%DELETE-I-FILDEL, $1$DJA16:<SYS0>DECNET.DIR;1 deleted (2 blocks)
   .
   .
   .
System root SYS1 deleted.

              Deleting directory tree SYS2...

%DELETE-I-FILDEL, $1$DJA16:<SYS1>DECNET.DIR;1 deleted (2 blocks)
   .
   .
   .
System root SYS2 deleted.

All the roots have been deleted.
%MOUNT-I-MOUNTED, JUPITR_SYS2  mounted on _$1$DJA16:

The second system disk has been created and mounted clusterwide.
Satellites can now be added.
```

## 8.6 Postconfiguration Tasks

Some configuration functions, such as adding or removing a voting member or enabling or disabling a quorum disk, require one or more additional operations.

These operations are listed in Table 8–10 and affect the integrity of the entire cluster. Follow the instructions in the table for the action you should take after executing either CLUSTER_CONFIG_LAN.COM or CLUSTER_CONFIG.COM to make major configuration changes.

**Table 8–10 Actions Required to Reconfigure a Cluster**

| After running the cluster configuration procedure to... | You should... |
|---|---|
| Add or remove a voting member | Update the AUTOGEN parameter files and the current system parameter files for all nodes in the cluster, as described in Section 8.6.1. |
| Enable a quorum disk | Perform the following steps:<br>1. Update the AUTOGEN parameter files and the current system parameter files for all quorum watchers in the cluster, as described in Section 8.6.1.<br>2. Reboot the nodes that have been enabled as quorum disk watchers (Section 2.3.8).<br>**Reference:** See also Section 8.2.4 for more information about adding a quorum disk. |

(continued on next page)

**Table 8–10 (Cont.) Actions Required to Reconfigure a Cluster**

| After running the cluster configuration procedure to... | You should... |
|---|---|
| Disable a quorum disk | Perform the following steps:<br>**Caution:** Do not perform these steps until you are ready to reboot the entire OpenVMS Cluster system. Because you are reducing quorum for the cluster, the votes cast by the quorum disk being removed could cause cluster partitioning.<br>1. Update the AUTOGEN parameter files and the current system parameter files for all quorum watchers in the cluster, as described in Section 8.6.1.<br>2. Evaluate whether or not quorum will be lost without the quorum disk:<br>**IF...** — **THEN...**<br>Quorum will not be lost — Perform these steps:<br>1. Use the DCL command SET CLUSTER/EXPECTED_VOTES to reduce the value of quorum.<br>2. Reboot the nodes that have been disabled as quorum disk watchers. (Quorum disk watchers are described in Section 2.3.8.)<br>Quorum will be lost — Shut down and reboot the entire cluster.<br>**Reference:** Cluster shutdown is described in Section 8.6.2.<br>**Reference:** See also Section 8.3.2 for more information about removing a quorum disk. |
| Add a satellite node | Perform these steps:<br>• Update the volatile network databases on other cluster members (Section 8.6.4).<br>• Optionally, alter the satellite's local disk label (Section 8.6.5). |
| Enable or disable the LAN for cluster communications | Update the current system parameter files and reboot the node on which you have enabled or disabled the LAN (Section 8.6.1). |

(continued on next page)

**Table 8–10 (Cont.) Actions Required to Reconfigure a Cluster**

| After running the cluster configuration procedure to... | You should... |
|---|---|
| Change allocation class values | Refer to the appropriate section, as follows:<br>• Change allocation class values on HSC subsystems (Section 6.2.2.2).<br>• Change allocation class values on HSJ subsystems (Section 6.2.2.3).<br>• Change allocation class values on DSSI ISE subsystems (Section 6.2.2.5).<br>• Update the current system parameter files and shut down and reboot the entire cluster (Sections 8.6.1 and 8.6.2). |
| Change the cluster group number or password | Shut down and reboot the entire cluster (Sections 8.6.2 and 8.6.7). |

### 8.6.1 Updating Parameter Files

The cluster configuration command procedures (CLUSTER_CONFIG_LAN.COM or CLUSTER_CONFIG.COM) can be used to modify parameters in the AUTOGEN parameter file for the node on which it is run.

In some cases, such as when you add or remove a voting cluster member, or when you enable or disable a quorum disk, you must update the AUTOGEN files for all the other cluster members.

Use either of the methods described in the following table.

| Method | Description |
|---|---|
| Update MODPARAMS.DAT files | Edit MODPARAMS.DAT in all cluster members' [SYS*x*.SYSEXE] directories and adjust the value for the EXPECTED_VOTES system parameter appropriately.<br>For example, if you add a voting member or if you enable a quorum disk, you must increment the value by the number of votes assigned to the new member (usually 1). If you add a voting member with one vote and enable a quorum disk with one vote on that computer, you must increment the value by 2. |
| Update AGEN$ files | Update the parameter settings in the appropriate AGEN$ include files:<br>• For satellites, edit SYS$MANAGER:AGEN$NEW_SATELLITE_DEFAULTS.DAT.<br>• For nonsatellites, edit SYS$MANAGER:AGEN$NEW_NODE_DEFAULTS.DAT.<br>**Reference:** These files are described in Section 8.2.2. |

You must also update the current system parameter files (VAXVMSSYS.PAR or ALPHAVMSSYS.PAR, as appropriate) so that the changes take effect on the next reboot.

Use either of the methods described in the following table.

| Method | Description |
|---|---|
| SYSMAN utility | Perform the following steps:<br>1. Log in as system manager.<br>2. Run the SYSMAN utility to update the EXPECTED_VOTES system parameter on all nodes in the cluster. For example:<br>`$ RUN SYS$SYSTEM:SYSMAN`<br>`%SYSMAN-I-ENV, current command environment:`<br>`  Clusterwide on local cluster`<br>`  Username SYSTEM  will be used on nonlocal nodes`<br>`SYSMAN> SET ENVIRONMENT/CLUSTER`<br>`SYSMAN> PARAM USE CURRENT`<br>`SYSMAN> PARAM SET EXPECTED_VOTES 2`<br>`SYSMAN> PARAM WRITE CURRENT`<br>`SYSMAN> EXIT` |
| AUTOGEN utility | Perform the following steps:<br>1. Log in as system manager.<br>2. Run the AUTOGEN utility to update the EXPECTED_VOTES system parameter on all nodes in the cluster. For example:<br>`$ RUN SYS$SYSTEM:SYSMAN`<br>`%SYSMAN-I-ENV, current command environment:`<br>`  Clusterwide on local cluster`<br>`  Username SYSTEM will be used on nonlocal nodes`<br>`SYSMAN> SET ENVIRONMENT/CLUSTER`<br>`SYSMAN> DO @SYS$UPDATE:AUTOGEN GETDATA SETPARAMS`<br>`SYSMAN> EXIT`<br>Do *not* specify the SHUTDOWN or REBOOT option.<br>**Hints:** If your next action is to shut down the node, you can specify SHUTDOWN or REBOOT (in place of SETPARAMS) in the DO @SYS$UPDATE:AUTOGEN GETDATA command. |

Both of these methods propagate the values to the computer's ALPHAVMSSYS.PAR file on Alpha computers or to the VAXVMSSYS.PAR file on VAX computers. In order for these changes to take effect, continue with the instructions in either Section 8.6.2 to shut down the cluster or in Section 8.6.3 to shut down the node.

## 8.6.2 Shutting Down the Cluster

Using the SYSMAN utility, you can shut down the entire cluster from a single node in the cluster. Follow these steps to perform an orderly shutdown:

1. Log in to the system manager's account on any node in the cluster.
2. Run the SYSMAN utility and specify the SET ENVIRONMENT/CLUSTER command. Be sure to specify the /CLUSTER_SHUTDOWN qualifier to the SHUTDOWN NODE command. For example:

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> SET ENVIRONMENT/CLUSTER
%SYSMAN-I-ENV, current command environment:
  Clusterwide on local cluster
  Username SYSTEM will be used on nonlocal nodes

SYSMAN> SHUTDOWN NODE/CLUSTER_SHUTDOWN/MINUTES_TO_SHUTDOWN=5 -
```

```
_SYSMAN> /AUTOMATIC_REBOOT/REASON="Cluster Reconfiguration"
%SYSMAN-I-SHUTDOWN, SHUTDOWN request sent to node
%SYSMAN-I-SHUTDOWN, SHUTDOWN request sent to node
SYSMAN>

SHUTDOWN message on JUPITR from user SYSTEM at JUPITR Batch   11:02:10
JUPITR will shut down in 5 minutes; back up shortly via automatic reboot.
Please log off node JUPITR.
Cluster Reconfiguration
SHUTDOWN message on JUPITR from user SYSTEM at JUPITR Batch   11:02:10
PLUTO will shut down in 5 minutes; back up shortly via automatic reboot.
Please log off node PLUTO.
Cluster Reconfiguration
```

For more information, see Section 10.7.

### 8.6.3 Shutting Down a Single Node

To stop a single node in an OpenVMS Cluster, you can use either the SYSMAN SHUTDOWN NODE command with the appropriate SET ENVIRONMENT command or the SHUTDOWN command procedure. These methods are described in the following table.

| Method | Description |
|---|---|
| SYSMAN utility | Follow these steps:<br>1. Log in to the system manager's account on any node in the OpenVMS Cluster.<br>2. Run the SYSMAN utility to shut down the node, as follows:<br>`$ RUN SYS$SYSTEM:SYSMAN`<br>`SYSMAN> SET ENVIRONMENT/NODE=JUPITR`<br>`Individual nodes: JUPITR`<br>`Username SYSTEM will be used on nonlocal nodes`<br>`SYSMAN> SHUTDOWN NODE/REASON="Maintenance" -`<br>`_SYSMAN> /MINUTES_TO_SHUTDOWN=5`<br>**Hint:** To shut down a subset of nodes in the cluster, you can enter several node names (separated by commas) on the SET ENVIRONMENT/NODE command. The following command shuts down nodes JUPITR and SATURN:<br>`SYSMAN> SET ENVIRONMENT/NODE=(JUPITR,SATURN)` |
| SHUTDOWN command procedure | Follow these steps:<br>1. Log in to the system manager's account on the node to be shut down.<br>2. Invoke the SHUTDOWN command procedure as follows:<br>`$ @SYS$SYSTEM:SHUTDOWN` |

For more information, see Section 10.7.

### 8.6.4 Updating Network Data

Whenever you add a satellite, the cluster configuration command procedure you use (CLUSTER_CONFIG_LAN.COM or CLUSTER_CONFIG.COM) updates both the permanent and volatile remote node network databases (NETNODE_REMOTE.DAT) on the boot server. However, the volatile databases on other cluster members are not automatically updated.

To share the new data throughout the cluster, you must update the volatile databases on all other cluster members. Log in as system manager, invoke the SYSMAN utility, and enter the following commands at the SYSMAN> prompt:

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> SET ENVIRONMENT/CLUSTER
%SYSMAN-I-ENV, current command environment:
        Clusterwide on local cluster
        Username SYSTEM        will be used on nonlocal nodes
SYSMAN> SET PROFILE/PRIVILEGES=(OPER,SYSPRV)
SYSMAN> DO MCR NCP SET KNOWN NODES ALL
%SYSMAN-I-OUTPUT, command execution on node X...
   .
   .
   .
SYSMAN> EXIT
$
```

The file NETNODE_REMOTE.DAT must be located in the directory SYS$COMMON:[SYSEXE].

### 8.6.5 Altering Satellite Local Disk Labels

If you want to alter the volume label on a satellite node's local page and swap disk, follow these steps after the satellite has been added to the cluster:

| Step | Action |
|---|---|
| 1 | Log in as system manager and enter a DCL command in the following format:<br>SET VOLUME/LABEL=volume-label device-spec[:]<br>**Note:** The SET VOLUME command requires write access (W) to the index file on the volume. If you are not the volume's owner, you must have either a system user identification code (UIC) or the SYSPRV privilege. |
| 2 | Update the [SYS*n*.SYSEXE]SATELLITE_PAGE.COM procedure on the boot server's system disk to reflect the new label. |

### 8.6.6 Changing Allocation Class Values

If you must change allocation class values on any HSC, HSJ, or DSSI ISE subsystem, you must do so while the entire cluster is shut down.

**Reference:** To change allocation class values:

- On computer systems, see Section 6.2.2.1.
- On HSC subsystems, see Section 6.2.2.2.
- On HSJ subsystems, see Section 6.2.2.3.
- On HSD subsystems, see Section 6.2.2.4.
- On DSSI subsystems, see Section 6.2.2.5.

### 8.6.7 Rebooting

The following table describes booting actions for satellite and storage subsystems:

| For configurations with... | You must... |
|---|---|
| HSC and HSJ subsystems | Reboot each computer after all HSC and HSJ subsystems have been set and rebooted. |
| Satellite nodes | Reboot boot servers before rebooting satellites.<br>Note that several new messages might appear. For example, if you have used the CLUSTER_CONFIG.COM CHANGE function to enable cluster communications over the LAN, one message reports that the LAN OpenVMS Cluster security database is being loaded.<br>**Reference:** See also Section 9.3 for more information about booting satellites. |
| DSSI ISE subsystems | Reboot the system after all the DSSI ISE subsystems have been set. |

For every disk-serving computer, a message reports that the MSCP server is being loaded.

To verify that all disks are being served in the manner in which you designed the configuration, at the system prompt ($) of the node serving the disks, enter the SHOW DEVICE/SERVED command. For example, the following display represents a DSSI configuration:

```
$ SHOW DEVICE/SERVED

Device:  Status  Total Size  Current  Max  Hosts
$1$DIA0   Avail     1954050        0    0      0
$1$DIA2   Avail     1800020        0    0      0
```

**Caution:** If you boot a node into an existing OpenVMS Cluster using minimum startup (the system parameter STARTUP_P1 is set to MIN), a number of processes (for example, CACHE_SERVER, CLUSTER_SERVER, and CONFIGURE) are not started. Compaq recommends that you start these processes manually if you intend to run the node in an OpenVMS Cluster system. Running a node without these processes enabled prevents the cluster from functioning properly.

**Reference:** Refer to the *OpenVMS System Manager's Manual* for more information about starting these processes manually.

### 8.6.8 Rebooting Satellites Configured with OpenVMS on a Local Disk

Satellite nodes can be set up to reboot automatically when recovering from system failures or power failures.

Reboot behavior varies from system to system. Many systems provide a console variable that allows you to specify which device to boot from by default. However, some systems have predefined boot "sniffers" that automatically detect a bootable device. The following table describes the rebooting conditions.

| IF... | AND... | THEN... |
|---|---|---|
| If your system does not allow you to specify the boot device for automatic reboot (that is, it has a boot sniffer) | An operating system is installed on the system's local disk | That disk will be booted in preference to requesting a satellite MOP load. To avoid this, you should take one of the measures in the following list before allowing any operation that causes an automatic reboot—for example, executing SYS$SYSTEM:SHUTDOWN.COM with the REBOOT option or using CLUSTER_CONFIG.COM to add that satellite to the cluster:<br>• Rename the directory file *ddcu*:[000000]SYS0.DIR on the local disk to *ddcu*:[000000]SYS*x*.DIR (where SYS*x* is a root other than SYS0, SYSE, or SYSF). Then enter the DCL command SET FILE/REMOVE as follows to remove the old directory entry for the boot image SYSBOOT.EXE:<br>`$ RENAME DUA0:[000000]SYS0.DIR DUA0:[000000]SYS1.DIR`<br>`$ SET FILE/REMOVE DUA0:[SYSEXE]SYSBOOT.EXE`<br>†On VAX systems, for subsequent reboots of VAX computers from the local disk, enter a command in the format B/*x*0000000 at the console-mode prompt (>>>). For example:<br>`>>> B/10000000`<br>• Disable the local disk. For instructions, refer to your computer-specific installation and operations guide. Note that this option is not available if the satellite's local disk is being used for paging and swapping. |

†VAX specific

## 8.7 Running AUTOGEN with Feedback

AUTOGEN includes a mechanism called **feedback**. This mechanism examines data collected during normal system operations, and it adjusts system parameters on the basis of the collected data whenever you run AUTOGEN with the feedback option. For example, the system records each instance of a disk server waiting for buffer space to process a disk request. Based on this information, AUTOGEN can size the disk server's buffer pool automatically to ensure that sufficient space is allocated.

Execute SYS$UPDATE:AUTOGEN.COM manually as described in the *OpenVMS System Manager's Manual*.

### 8.7.1 Advantages

To ensure that computers are configured adequately when they first join the cluster, you can run AUTOGEN with feedback automatically as part of the initial boot sequence. Although this step adds an additional reboot before the computer can be used, the computer's performance can be substantially improved.

Compaq strongly recommends that you use the feedback option. Without feedback, it is difficult for AUTOGEN to anticipate patterns of resource usage, particularly in complex configurations. Factors such as the number of computers and disks in the cluster and the types of applications being run require adjustment of system parameters for optimal performance.

Compaq also recommends using AUTOGEN with feedback rather than the SYSGEN utility to modify system parameters, because AUTOGEN:

- Uses parameter changes in MODPARAMS.DAT and AGEN$ files. (Changes recorded in MODPARAMS.DAT are not lost during updates to the OpenVMS operating system.)
- Reconfigures other system parameters to reflect changes.

### 8.7.2 Initial Values

When a computer is first added to an OpenVMS Cluster, system parameters that control the computer's system resources are normally adjusted in several steps, as follows:

1. The cluster configuration command procedure (CLUSTER_CONFIG_LAN.COM or CLUSTER_CONFIG.COM) sets initial parameters that are adequate to boot the computer in a minimum environment.
2. When the computer boots, AUTOGEN runs automatically to size the static operating system (without using any dynamic feedback data), and the computer reboots into the OpenVMS Cluster environment.
3. After the newly added computer has been subjected to typical use for a day or more, you should run AUTOGEN with feedback manually to adjust parameters for the OpenVMS Cluster environment.
4. At regular intervals, and whenever a major change occurs in the cluster configuration or production environment, you should run AUTOGEN with feedback manually to readjust parameters for the changes.

Because the first AUTOGEN operation (initiated by either CLUSTER_CONFIG_LAN.COM or CLUSTER_CONFIG.COM) is performed both in the minimum environment and without feedback, a newly added computer may be inadequately configured to run in the OpenVMS Cluster environment. For this reason, you might want to implement additional configuration measures like those described in Section 8.7.3 and Section 8.7.4.

### 8.7.3 Obtaining Reasonable Feedback

When a computer first boots into an OpenVMS Cluster, much of the computer's resource utilization is determined by the current OpenVMS Cluster configuration. Factors such as the number of computers, the number of disk servers, and the number of disks available or mounted contribute to a fixed minimum resource requirements. Because this minimum does not change with continued use of the computer, feedback information about the required resources is immediately valid.

Other feedback information, however, such as that influenced by normal user activity, is not immediately available, because the only "user" has been the system startup process. If AUTOGEN were run with feedback at this point, some system values might be set too low.

By running a simulated user load at the end of the first production boot, you can ensure that AUTOGEN has reasonable feedback information. The User Environment Test Package (UETP) supplied with your operating system contains a test that simulates such a load. You can run this test (the UETP LOAD phase) as part of the initial production boot, and then run AUTOGEN with feedback before a user is allowed to log in.

To implement this technique, you can create a command file like that in step 1 of the procedure in Section 8.7.4, and submit the file to the computer's local batch queue from the cluster common SYSTARTUP procedure. Your command file conditionally runs the UETP LOAD phase and then reboots the computer with AUTOGEN feedback.

### 8.7.4 Creating a Command File to Run AUTOGEN

As shown in the following sample file, UETP lets you specify a typical user load to be run on the computer when it first joins the cluster. The UETP run generates data that AUTOGEN uses to set appropriate system parameter values for the computer when rebooting it with feedback. Note, however, that the default setting for the UETP user load assumes that the computer is used as a timesharing system. This calculation can produce system parameter values that might be excessive for a single-user workstation, especially if the workstation has large memory resources. Therefore, you might want to modify the default user load setting, as shown in the sample file.

Follow these steps:

1. Create a command file like the following:

```
$!
$!   ***** SYS$COMMON:[SYSMGR]UETP_AUTOGEN.COM *****
$!
$! For initial boot only, run UETP LOAD phase and
$! reboot with AUTOGEN feedback.
$!
$ SET NOON
$ SET PROCESS/PRIVILEGES=ALL
$!
$! Run UETP to simulate a user load for a satellite
$! with 8 simultaneously active user processes. For a
$! CI connected computer, allow UETP to calculate the load.
$!
$ LOADS = "8"
$ IF F$GETDVI("PAA0:","EXISTS") THEN LOADS = ""
$ @UETP LOAD 1 'loads'
$!
$! Create a marker file to prevent resubmission of
$! UETP_AUTOGEN.COM at subsequent reboots.
$!
$ CREATE SYS$SPECIFIC:[SYSMGR]UETP_AUTOGEN.DONE
$!
$! Reboot with AUTOGEN to set SYSGEN values.
$!
$ @SYS$UPDATE:AUTOGEN SAVPARAMS REBOOT FEEDBACK
$!
$ EXIT
```

2. Edit the cluster common SYSTARTUP file and add the following commands at the end of the file. Assume that queues have been started and that a batch queue is running on the newly added computer. Submit UETP_AUTOGEN.COM to the computer's local batch queue.

```
$!
$ NODE = F$GETSYI("NODE")
$ IF F$SEARCH ("SYS$SPECIFIC:[SYSMGR]UETP_AUTOGEN.DONE") .EQS. ""
$ THEN
$ SUBMIT /NOPRINT /NOTIFY /USERNAME=SYSTEST -
_$ /QUEUE='NODE'_BATCH SYS$MANAGER:UETP_AUTOGEN
```

```
$ WAIT_FOR_UETP:
$  WRITE SYS$OUTPUT "Waiting for UETP and AUTOGEN... ''F$TIME()'"
$  WAIT 00:05:00.00             ! Wait 5 minutes
$  GOTO WAIT_FOR_UETP
$ ENDIF
$!
```

**Note:** UETP must be run under the user name SYSTEST.

3. Execute CLUSTER_CONFIG_LAN.COM or CLUSTER_CONFIG.COM to add the computer.

When you boot the computer, it runs UETP_AUTOGEN.COM to simulate the user load you have specified, and it then reboots with AUTOGEN feedback to set appropriate system parameter values.

# 9
# Building Large OpenVMS Cluster Systems

This chapter provides guidelines for building OpenVMS Cluster systems that include many computers—approximately 20 or more—and describes procedures that you might find helpful. (Refer to the OpenVMS Cluster Software *Software Product Description* (SPD) for configuration limitations.) Typically, such OpenVMS Cluster systems include a large number of satellites.

Note that the recommendations in this chapter also can prove beneficial in some clusters with fewer than 20 computers. Areas of discussion include:

- Booting
- Availability of MOP and disk servers
- Multiple system disks
- Shared resource availability
- Hot system files
- System disk space
- System parameters
- Network problems
- Cluster alias

## 9.1 Setting Up the Cluster

When building a new large cluster, you must be prepared to run AUTOGEN and reboot the cluster several times during the installation. The parameters that AUTOGEN sets for the first computers added to the cluster will probably be inadequate when additional computers are added. Readjustment of parameters is critical for boot and disk servers.

One solution to this problem is to run the UETP_AUTOGEN.COM command procedure (described in Section 8.7.4) to reboot computers at regular intervals as new computers or storage interconnects are added. For example, each time there is a 10% increase in the number of computers, storage, or interconnects, you should run UETP_AUTOGEN.COM. For best results, the last time you run the procedure should be as close as possible to the final OpenVMS Cluster environment.

To set up a new, large OpenVMS Cluster, follow these steps:

| Step | Task |
|---|---|
| 1 | Configure boot and disk servers using the CLUSTER_CONFIG_LAN.COM or the CLUSTER_CONFIG.COM command procedure (described in Chapter 8). |
| 2 | Install all layered products and site-specific applications required for the OpenVMS Cluster environment, or as many as possible. |
| 3 | Prepare the cluster startup procedures so that they are as close as possible to those that will be used in the final OpenVMS Cluster environment. |
| 4 | Add a small number of satellites (perhaps two or three) using the cluster configuration command procedure. |
| 5 | Reboot the cluster to verify that the startup procedures work as expected. |
| 6 | After you have verified that startup procedures work, run UETP_AUTOGEN.COM on every computer's local batch queue to reboot the cluster again and to set initial production environment values. When the cluster has rebooted, all computers should have reasonable parameter settings. However, check the settings to be sure. |
| 7 | Add additional satellites to double their number. Then rerun UETP_AUTOGEN on each computer's local batch queue to reboot the cluster, and set values appropriately to accommodate the newly added satellites. |
| 8 | Repeat the previous step until all satellites have been added. |
| 9 | When all satellites have been added, run UETP_AUTOGEN a final time on each computer's local batch queue to reboot the cluster and to set new values for the production environment. |

For best performance, do not run UETP_AUTOGEN on every computer simultaneously, because the procedure simulates a user load that is probably more demanding than that for the final production environment. A better method is to run UETP_AUTOGEN on several satellites (those with the least recently adjusted parameters) while adding new computers. This technique increases efficiency because little is gained when a satellite reruns AUTOGEN shortly after joining the cluster.

For example, if the entire cluster is rebooted after 30 satellites have been added, few adjustments are made to system parameter values for the 28th satellite added, because only two satellites have joined the cluster since that satellite ran UETP_AUTOGEN as part of its initial configuration.

## 9.2 General Booting Considerations

Two general booting considerations, concurrent booting and minimizing boot time, are described in this section.

### 9.2.1 Concurrent Booting

One of the rare times when all OpenVMS Cluster computers are simultaneously active is during a cluster reboot—for example, after a power failure. All satellites are waiting to reload the operating system, and as soon as a boot server is available, they begin to boot in parallel. This booting activity places a significant I/O load on the system disk or disks, interconnects, and boot servers.

For example, Table 9–1 shows a VAX system disk's I/O activity and elapsed time until login for a single satellite with minimal startup procedures when the satellite is the only one booting. Table 9–2 shows system disk I/O activity and time elapsed between boot server response and login for various numbers of satellites booting from a single system disk. The disk in these examples has a capacity of 40 I/O operations per second.

Note that the numbers in the tables are fabricated and are meant to provide only a generalized picture of booting activity. Elapsed time until login on satellites in any particular cluster depends on the complexity of the site-specific system startup procedures. Computers in clusters with many layered products or site-specific applications require more system disk I/O operations to complete booting operations.

**Table 9–1 Sample System Disk I/O Activity and Boot Time for a Single VAX Satellite**

| Total I/O Requests to System Disk | Average System Disk I/O Operations per Second | Elapsed Time Until Login (minutes) |
|---|---|---|
| 4200 | 6 | 12 |

**Table 9–2 Sample System Disk I/O Activity and Boot Times for Multiple VAX Satellites**

| Number of Satellites | I/Os Requested per Second | I/Os Serviced per Second | Elapsed Time Until Login (minutes) |
|---|---|---|---|
| 1 | 6 | 6 | 12 |
| 2 | 12 | 12 | 12 |
| 4 | 24 | 24 | 12 |
| 6 | 36 | 36 | 12 |
| 8 | 48 | 40 | 14 |
| 12 | 72 | 40 | 21 |
| 16 | 96 | 40 | 28 |
| 24 | 144 | 40 | 42 |
| 32 | 192 | 40 | 56 |
| 48 | 288 | 40 | 84 |
| 64 | 384 | 40 | 112 |
| 96 | 576 | 40 | 168 |

While the elapsed times shown in Table 9–2 do not include the time required for the boot server itself to reload, they illustrate that the I/O capacity of a single system disk can be the limiting factor for cluster reboot time.

### 9.2.2 Minimizing Boot Time

A large cluster needs to be carefully configured so that there is sufficient capacity to boot the desired number of nodes in the desired amount of time. As shown in Table 9–2, the effect of 96 satellites rebooting could induce an I/O bottleneck that can stretch the OpenVMS Cluster reboot times into hours. The following list provides a few methods to minimize boot times.

- Careful configuration techniques

  *Guidelines for OpenVMS Cluster Configurations* contains data on configurations and the capacity of the computers, system disks, and interconnects involved.

- Adequate system disk throughput

  Achieving enough system disk throughput typically requires a combination of techniques. Refer to Section 9.5 for complete information.

- Sufficient network bandwidth

  A single Ethernet is unlikely to have sufficient bandwidth to meet the needs of a large OpenVMS Cluster. Likewise, a single Ethernet adapter may become a bottleneck, especially for a disk server. Sufficient network bandwidth can be provided using some of the techniques listed in step 1 of Table 9–3.

- Installation of only the required layered products and devices.

## 9.3 Booting Satellites

OpenVMS Cluster satellite nodes use a single LAN adapter for the initial stages of booting. If a satellite is configured with multiple LAN adapters, the system manager can specify with the console BOOT command which adapter to use for the initial stages of booting. Once the system is running, the OpenVMS Cluster uses all available LAN adapters. This flexibility allows you to work around broken adapters or network problems.

The procedures and utilities for configuring and booting satellite nodes are the same or vary only slightly between Alpha and VAX systems. These are described in Section 9.4.

In addition, VAX nodes can MOP load Alpha satellites, and Alpha nodes can MOP load VAX satellites. Cross-architecture booting is described in Section 10.5.

## 9.4 Configuring and Booting Satellite Nodes

Complete the items in the following Table 9–3 before proceeding with satellite booting.

**Table 9–3 Checklist for Satellite Booting**

| Step | Action |
|---|---|
| **1** | Configure disk server LAN adapters.<br>Because disk-serving activity in an OpenVMS Cluster system can generate a substantial amount of I/O traffic on the LAN, boot and disk servers should use the highest-bandwidth LAN adapters in the cluster. The servers can also use multiple LAN adapters in a single system to distribute the load across the LAN adapters.<br>The following list suggests ways to provide sufficient network bandwidth:<br>– Select network adapters with sufficient bandwidth.<br>– Use switches to segregate traffic and to provide increased total bandwidth.<br>– Use multiple LAN adapters on MOP and disk servers.<br>– Use switch or higher speed LAN, fanning out to slower LAN segments.<br>– Use multiple independent networks.<br>– Provide sufficient MOP and disk server CPU capacity by selecting a computer with sufficient power and by configuring multiple server nodes to share the load. |
| **2** | If the MOP server node and system-disk server node (Alpha or VAX) are not already configured as cluster members, follow the directions in Section 8.4 for using the cluster configuration command procedure to configure each of the VAX or Alpha nodes. Include multiple boot and disk servers to enhance availability and distribute I/O traffic over several cluster nodes. |

(continued on next page)

**Table 9–3 (Cont.) Checklist for Satellite Booting**

| Step | Action |
|---|---|
| **3** | Configure additional memory for disk serving. |
| **4** | Run the cluster configuration procedure on the Alpha or VAX node for each satellite you want to boot into the OpenVMS Cluster. |

### 9.4.1 Booting from a Single LAN Adapter

To boot a satellite, enter the following command:

```
>>> BOOT LAN-adapter-device-name
```

In the example, the *LAN-adapter-device-name* could be any valid LAN adapter name, for example EZA0 or XQB0.

If you need to perform a conversational boot, use the command shown in the following table.

| IF the system is... | THEN... |
|---|---|
| An Alpha system | At the Alpha system console prompt (>>>), enter:<br>`>>> b -flags 0,1 eza0`<br>In this example, -flags stands for the flags command line qualifier, which takes two values:<br>• System root number<br>The "0" tells the console to boot from the system root [SYS0]. This is ignored when booting satellite nodes because the system root comes from the network database of the boot node.<br>• Conversational boot flag<br>The "1" indicates that the boot should be conversational.<br>The argument eza0 is the LAN adapter to be used for booting.<br>Finally, notice that a load file is not specified in this boot command line. For satellite booting, the load file is part of the node description in the DECnet or LANCP database. |
| A VAX system | At the VAX system console prompt (>>>), specify the full device name in the boot command:<br>`>>>B/R5=1 XQB0`<br>The exact syntax will vary depending on system type. Please refer to the hardware user's guide for your system. |

If the boot fails:

- If the configuration permits and the network database is properly set up, reenter the boot command using another LAN adapter (see Section 9.4.4).
- See Section C.3.5 for information about troubleshooting satellite booting problems.

### 9.4.2 Changing the Default Boot Adapter

To change the default boot adapter, you need the physical address of the alternate LAN adapter. You use the address to update the satellite's node definition in the DECnet or LANCP database on the MOP servers so that they recognize the satellite (described in Section 9.4.4).

| IF the system is... | THEN... |
|---|---|
| An Alpha system | Use the SHOW CONFIG console command to find the LAN address of additional adapters. |
| A VAX system | Use the following method to find the LAN address of additional adapters:<br>• Enter the console command SHOW ETHERNET.<br>• Boot the READ_ADDR program using the following commands:<br>`>>>B/100 XQB0`<br>`Bootfile:READ_ADDR` |

### 9.4.3 Booting from Multiple LAN Adapters (Alpha Only)

On Alpha systems, availability can be increased by using multiple LAN adapters for booting because access to the MOP server and disk server can occur via different LAN adapters. To use multiple adapter booting, perform the steps in the following table.

| Step | Task |
|---|---|
| **1** | Obtain the physical addresses of the additional LAN adapters. |
| **2** | Use these addresses to update the node definition in the DECnet or LANCP database on some of the MOP servers so that they recognize the satellite (described in Section 9.4.4). |
| **3** | If the satellite is already defined in the DECnet database, skip to step 4. If the satellite is not defined in the DECnet database, specify the SYS$SYSTEM:APB.EXE downline load file in the Alpha network database (see Example 10–2). |
| **4** | Specify multiple LAN adapters on the boot command line. (Use the SHOW DEVICE or SHOW CONFIG console command to obtain the names of adapters.) |

The following command line is the same as that used for booting from a single LAN adapter on an Alpha system (see Section 9.4.2) except that it lists two LAN adapters, eza0 and ezb0, as the devices from which to boot:

```
>>> b -flags 0,1 eza0, ezb0
```

In this command line:

| Stage | What Happens |
|---|---|
| **1** | MOP booting is attempted from the first device (eza0). If that fails, MOP booting is attempted from the next device (ezb0). When booting from network devices, if the MOP boot attempt fails from all devices, then the console starts again from the first device. |
| **2** | Once the MOP load has completed, the boot driver starts the NISCA protocol on all of the LAN adapters. The NISCA protocol is used to access the system disk server and finish loading the operating system (see Appendix F). |

### 9.4.4 Enabling Satellites to Use Alternate LAN Adapters for Booting

OpenVMS supports only one hardware address attribute per remote node definition in either a DECnet or LANCP database. To enable a satellite with multiple LAN adapters to use any LAN adapter to boot into the cluster, two different methods are available:

- Define a pseudonode for each additional LAN adapter.
- Create and maintain different node databases for different boot nodes.

#### Defining Pseudonodes for Additional LAN Adapters

When defining a pseudonode with a different DECnet or LANCP address:

- Make sure the address points to the same cluster satellite root directory as the existing node definition (to associate the pseudonode with the satellite).
- Specify the hardware address of the alternate LAN adapter in the pseudonode definition.

For DECnet, follow the procedure shown in Table 9–4. For LANCP, follow the procedure shown in Table 9–5.

**Table 9–4 Procedure for Defining a Pseudonode Using DECnet MOP Services**

| Step | Procedure | Comments |
|---|---|---|
| 1 | Display the node's existing definition using the following NCP command:<br><br>`$ RUN SYS$SYSTEM:NCP`<br>`NCP> SHOW NODE` *node-name* `CHARACTERISTICS` | This command displays a list of the satellite's characteristics, such as its hardware address, load assist agent, load assist parameter, and more. |
| 2 | Create a pseudonode by defining a unique DECnet address and node name at the NCP command prompt, as follows:<br><br>‡`DEFINE NODE` *pseudo-area.pseudo-number* `-`<br>`NAME` *pseudo-node-name* `-`<br>`LOAD FILE APB.EXE -`<br>`LOAD ASSIST AGENT SYS$SHARE:NISCS_LAA.EXE -`<br>`LOAD ASSIST PARAMETER` *disk$sys*`:[<`*root*`.>] -`<br>`HARDWARE ADDRESS` *xx-xx-xx-xx-xx-xx* | This example is specific to an Alpha node. For a VAX node, replace the command LOAD FILE APB.EXE, with TERTIARY LOADER SYS$SYSTEM:TERTIARY_VMB.EXE. |

‡Alpha specific

**Table 9–5 Procedure for Defining a Pseudonode Using LANCP MOP Services**

| Step | Procedure | Comments |
|---|---|---|
| 1 | Display the node's existing definition using the following LANCP command:<br><br>`$ RUN SYS$SYSTEM:LANCP`<br>`LANCP> SHOW NODE` *node-name* | This command displays a list of the satellite's characteristics, such as its hardware address and root directory address. |

(continued on next page)

**Table 9–5 (Cont.) Procedure for Defining a Pseudonode Using LANCP MOP Services**

| Step | Procedure | Comments |
|---|---|---|
| **2** | Create a pseudonode by defining a unique LANCP address and node name at the LANCP command prompt, as follows:<br><br>‡DEFINE NODE *pseudo-node-name* -<br>/FILE= APB.EXE -<br>/ROOT=*disk$sys*:[<*root*.>] -<br>/ADDRESS=*xx-xx-xx-xx-xx-xx* | This example is specific to an Alpha node. For a VAX node, replace the qualifier FILE=APB.EXE with FILE=NISCS_LOAD.EXE. |

‡Alpha specific

### Creating Different Node Databases for Different Boot Nodes

When creating different DECnet or LANCP databases on different boot nodes:

- Set up the databases so that a system booting from one LAN adapter receives responses from a subset of the MOP servers. The same system booting from a different LAN adapter receives responses from a different subset of the MOP servers.
- In each database, list a different LAN address for the same node definition.

The procedures are similar for DECnet and LANCP, but the database file names, utilities, and commands differ. For the DECnet procedure, see Table 9–6. For the LANCP procedure, see Table 9–7.

**Table 9–6 Procedure for Creating Different DECnet Node Databases**

| Step | Procedure | Comments |
|---|---|---|
| **1** | Define the logical name NETNODE_REMOTE to different values on different nodes so that it points to different files. | The logical NETNODE_REMOTE points to the working copy of the remote node file you are creating. |
| **2** | Locate NETNODE_REMOTE.DAT files in the system-specific area for each node.<br><br>On each of the various boot servers, ensure that the hardware address is defined as a unique address that matches one of the adapters on the satellite. Enter the following commands at the NCP command prompt:<br><br>‡DEFINE NODE *area.number* -<br>NAME *node-name* -<br>LOAD FILE APB.EXE -<br>LOAD ASSIST AGENT SYS$SHARE:NISCS_LAA.EXE -<br>LOAD ASSIST PARAMETER *disk$sys*:[<*root*.>] -<br>HARDWARE ADDRESS *xx-xx-xx-xx-xx-xx* | A NETNODE_REMOTE.DAT file located in [SYS0.SYSEXE] overrides one located in [SYS0.SYSCOMMON.SYSEXE] for a system booting from system root 0.<br><br>If the NETNODE_REMOTE.DAT files are copies of each other, the node name, tertiary loader (for VAX) or LOAD FILE (for Alpha), load assist agent, and load assist parameter are already set up. You need only specify the new hardware address.<br><br>Because the default hardware address is stored in NETUPDATE.COM, you must also edit this file on the second boot server. |

‡Alpha specific

**Table 9–7 Procedure for Creating Different LANCP Node Databases**

| Step | Procedure | Comments |
|---|---|---|
| **1** | Define the logical name LAN$NODE_DATABASE to different values on different nodes so that it points to different files. | The logical LAN$NODE_DATABASE points to the working copy of the remote node file you are creating. |

(continued on next page)

**Table 9–7 (Cont.) Procedure for Creating Different LANCP Node Databases**

| Step | Procedure | Comments |
|---|---|---|
| **2** | Locate LAN$NODE_DATABASE.DAT files in the system-specific area for each node.<br><br>On each of the various boot servers, ensure that the hardware address is defined as a unique address that matches one of the adapters on the satellite. Enter the following commands at the LANCP command prompt:<br><br>‡DEFINE NODE *node-name* -<br>/FILE= APB.EXE -<br>/ROOT=*disk$sys*:[<*root*.>] -<br>/ADDRESS=*xx-xx-xx-xx-xx-xx* | If the LAN$NODE_DATABASE.DAT files are copies of each other, the node name and the FILE and ROOT qualifier values are already set up. You need only specify the new address. |

‡Alpha specific

Once the satellite receives the MOP downline load from the MOP server, the satellite uses the booting LAN adapter to connect to any node serving the system disk. The satellite continues to use the LAN adapters on the boot command line exclusively until after the run-time drivers are loaded. The satellite then switches to using the run-time drivers and starts the local area OpenVMS Cluster protocol on all of the LAN adapters.

For additional information about the NCP command syntax, refer to *DECnet for OpenVMS Network Management Utilities*.

**For DECnet–Plus:** On an OpenVMS Cluster running DECnet–Plus, you do not need to take the same actions in order to support a satellite with more than one LAN adapter. The DECnet–Plus support to downline load a satellite allows for an entry in the database that contains a list of LAN adapter addresses. See the DECnet–Plus documentation for complete information.

### 9.4.5 Configuring MOP Service

On a boot node, CLUSTER_CONFIG.COM enables the DECnet MOP downline load service on the first circuit that is found in the DECnet database.

On systems running DECnet for OpenVMS, display the circuit state and the service (MOP downline load service) state using the following command:

```
$ MCR NCP SHOW CHAR KNOWN CIRCUITS

          .
          .
          .
   Circuit = SVA-0

   State                    = on
   Service                  = enabled
          .
          .
          .
```

This example shows that circuit SVA-0 is in the ON state with the MOP downline service enabled. This is the correct state to support MOP downline loading for satellites.

Enabling MOP service on additional LAN adapters (circuits) must be performed manually. For example, enter the following NCP commands to enable service for the circuit QNA-1:

```
$ MCR NCP SET CIRCUIT QNA-1 STATE OFF
$ MCR NCP SET CIRCUIT QNA-1 SERVICE ENABLED STATE ON
$ MCR NCP DEFINE CIRCUIT QNA-1 SERVICE ENABLED
```

**Reference:** For more details, refer to *DECnet-Plus for OpenVMS Network Management*.

### 9.4.6 Controlling Satellite Booting

You can control the satellite boot process in a number of ways. Table 9–8 shows examples specific to DECnet for OpenVMS. Refer to the DECnet–Plus documentation for equivalent information.

**Table 9–8 Controlling Satellite Booting**

| Method | Comments |
|---|---|
| **Use DECbootsync** | |
| To control the number of workstations starting up simulataneously, use DECbootsync, which is available through the NSIS Reusable Software library: `http://eadc.aeo.dec.com/`. DECbootsync uses the distributed lock manager to control the number of satellites allowed to continue their startup command procedures at the same time. | DECbootsync does not control booting of the OpenVMS operating system, but controls the execution of startup command procedures and installation of layered products on satellites. |
| **Disable MOP service on MOP servers temporarily** | |
| Until the MOP server can complete its own startup operations, boot requests can be temporarily disabled by setting the DECnet Ethernet circuit to a "Service Disabled" state as shown:<br>**1** To disable MOP service during startup of a MOP server, enter the following commands:<br>`$ MCR NCP DEFINE CIRCUIT MNA-1 -`<br>`_$ SERVICE DISABLED`<br>`$ @SYS$MANAGER:STARTNET`<br>`$ MCR NCP DEFINE CIRCUIT MNA-1 -`<br>`_$ SERVICE ENABLED`<br>**2** To reenable MOP service later, enter the following commands in a command procedure so that they execute quickly and so that DECnet service to the users is not disrupted:<br>`$ MCR NCP`<br>`NCP> SET CIRCUIT MNA-1 STATE OFF`<br>`NCP> SET CIRCUIT MNA-1 SERVICE ENABLED`<br>`NCP> SET CIRCUIT MNA-1 STATE ON` | This method prevents the MOP server from servicing the satellites; it does not prevent the satellites from requesting a boot from other MOP servers.<br>If a satellite that is requesting a boot receives no response, it will make fewer boot requests over time. Thus, booting the satellite may take longer than normal once MOP service is reenabled.<br>1. MNA-1 represents the MOP service circuit.<br>After entering these commands, service will be disabled in the volatile database. Do not disable service permanently.<br>2. Reenable service as shown. |

(continued on next page)

**Table 9–8 (Cont.) Controlling Satellite Booting**

| Method | Comments |
|---|---|
| **Disable MOP service for individual satellites** | |
| You can disable requests temporarily on a per-node basis in order to clear a node's information from the DECnet database. Clear a node's information from DECnet database on the MOP server using NCP, then reenable nodes as desired to control booting:<br>**1** To disable MOP service for a given node, enter the following command:<br>`$ MCR NCP`<br>`NCP> CLEAR NODE *satellite* HARDWARE ADDRESS`<br>**2** To reenable MOP service for that node, enter the following command:<br>`$ MCR NCP`<br>`NCP> SET NODE *satellite* ALL` | This method does not prevent satellites from requesting boot service from another MOP server.<br>1. After entering the commands, service will be disabled in the volatile database. Do not disable service permanently.<br>2. Reenable service as shown. |

(continued on next page)

**Table 9–8 (Cont.) Controlling Satellite Booting**

| Method | Comments |
|---|---|
| **Bring satellites to console prompt on shutdown** | |
| Use any of the following methods to halt a satellite so that it halts (rather than reboots) upon restoration of power.<br>**1** Use the VAXcluster Console System (VCS).<br>**2** Stop in console mode upon Halt or powerup:<br>For Alpha computers:<br>`>>> (SET AUTO_ACTION HALT)`<br>For VAX 3100 or VAX 4000 series computers:<br>`>>> (SET HALT 3)`<br>For VAX 2000 series computers:<br>`>>> TEST 53`<br>`2 ? >>> 3`<br>**3** Set up a satellite so that it will stop in console mode when a HALT instruction is executed according to the instructions in the following list.<br>a. Enter the following NCP commands so that a reboot will load an image that does a HALT instruction:<br>`$ MCR NCP`<br>`NCP> CLEAR NODE node LOAD ASSIST PARAMETER`<br>`NCP> CLEAR NODE node LOAD ASSIST AGENT`<br>`NCP> SET NODE node LOAD FILE -`<br>`_ MOM$LOAD:READ_ADDR.SYS`<br>b. Shut down the satellite, and specify an immediate reboot using the following SYSMAN command:<br>`$ MCR SYSMAN`<br>`SYSMAN> SET ENVIRONMENT/NODE=satellite`<br>`SYSMAN> DO @SYS$UPDATE:AUTOGEN REBOOT`<br>c. When you want to allow the satellite to boot normally, enter the following NCP commands so that OpenVMS will be loaded later:<br>`$ MCR NCP`<br>`NCP> SET NODE satellite ALL` | If you plan to use the DECnet Trigger operation, it is important to use a program to perform a HALT instruction that causes the satellite to enter console mode. This is because systems that support remote triggering only support it while the system is in console mode.<br>1. Some, but not all, satellites can be set up so they halt upon restoration of power or execution of a HALT instruction rather than automatically rebooting.<br>**Note:** You need to enter the SET commands only once on each system because the settings are saved in nonvolatile RAM.<br>2. The READ_ADDR.SYS program, which is normally used to find out the Ethernet address of a satellite node, also executes a HALT instruction upon its completion. |

(continued on next page)

**Table 9–8 (Cont.) Controlling Satellite Booting**

| Method | Comments |
|---|---|
| **Boot satellites remotely with Trigger** | |
| The console firmware in some satellites, such as the VAX 3100 and VAX 4000, allow you to boot them remotely using the DECnet Trigger operation. You must turn on this capability at the console prompt before you enter the NCP command TRIGGER, as shown:<br>**1** To boot VAX satellites using the DECnet Trigger facility, enter these commands at the console prompt:<br>`>>> SET MOP 1`<br>`>>> SET TRIGGER 1`<br>`>>> SET PSWD`<br>**2** Trigger a satellite boot remotely by entering the following commands at the NCP prompt, as follows:<br>`$ MCR NCP`<br>`NCP> TRIGGER NODE` *satellite* `-`<br>`_ VIA MNA-1 SERVICE PASSWORD` *password* | Optionally, you can set up the MOP server to run a command procedure and trigger 5 or 10 satellites at a time to stagger the boot-time work load. You can boot satellites in a priority order, for example, first boot your satellite, then high-priority satellites, and so on.<br>1. The SET TRIGGER 1 command enables the DECnet MOP listener, and the SET PSWD command enables remote triggering. The SET PSWD command prompts you twice for a 16-digit hexadecimal password string that is used to validate a remote trigger request.<br>**Note:** You need to enter the SET commands only once on each system, because the settings are saved in nonvolatile RAM.<br>2. MNA-1 represents the MOP service circuit, and *password* is the hexadecimal number that you specified in step 1 with the SET PSWD command. |

**Important:** When the SET HALT command is set up as described in Table 9–8, a power failure will cause the satellite to stop at the console prompt instead of automatically rebooting when power is restored. This is appropriate for a mass power failure, but if someone trips over the power cord for a single satellite it can result in unnecessary unavailability.

You can provide a way to scan and trigger a reboot of satellites that go down this way by simply running a batch job periodically that performs the following tasks:

1. Uses the DCL lexical function F$GETSYI to check each node that should be in the cluster.
2. Checks the CLUSTER_MEMBER lexical item.
3. Issues an NCP TRIGGER command for any satellite that is not currently a member of the cluster.

## 9.5 System-Disk Throughput

Achieving enough system-disk throughput requires some combination of the following techniques:

| Technique | Reference |
|---|---|
| Avoid disk rebuilds at boot time. | Section 9.5.1 |
| Offload work from the system disk. | Section 9.5.2 |
| Configure multiple system disks. | Section 9.5.3 |
| Use Volume Shadowing for OpenVMS. | Section 6.6 |

### 9.5.1 Avoiding Disk Rebuilds

The OpenVMS file system maintains a cache of preallocated file headers and disk blocks. When a disk is not properly dismounted, such as when a system fails, this preallocated space becomes temporarily unavailable. When the disk is mounted again, OpenVMS scans the disk to recover that space. This is called a **disk rebuild**.

A large OpenVMS Cluster system must ensure sufficient capacity to boot nodes in a reasonable amount of time. To minimize the impact of disk rebuilds at boot time, consider making the following changes:

| Action | Result |
|---|---|
| Use the DCL command MOUNT/NOREBUILD for all user disks, at least on the satellite nodes. Enter this command into startup procedures that mount user disks. | It is undesirable to have a satellite node rebuild the disk, yet this is likely to happen if a satellite is the first to reboot after it or another node fails. |
| Set the system parameter ACP_REBLDSYSD to 0, at least for the satellite nodes. | This prevents a rebuild operation on the system disk when it is mounted implicitly by OpenVMS early in the boot process. |
| Avoid a disk rebuild during prime working hours by using the SET VOLUME/REBUILD command during times when the system is not so heavily used. Once the computer is running, you can run a batch job or a command procedure to execute the SET VOLUME/REBUILD command for each disk drive. | User response times can be degraded during a disk rebuild operation because most I/O activity on that disk is blocked. Because the SET VOLUME/REBUILD command determines whether a rebuild is needed, the job can execute the command for every disk. This job can be run during off hours, preferably on one of the more powerful nodes. |

**Caution:** In large OpenVMS Cluster systems, large amounts of disk space can be preallocated to caches. If many nodes abruptly leave the cluster (for example, during a power failure), this space becomes temporarily unavailable. If your system usually runs with nearly full disks, do not disable rebuilds on the server nodes at boot time.

### 9.5.2 Offloading Work

In addition to the system disk throughput issues during an entire OpenVMS Cluster boot, access to particular system files even during steady-state operations (such as logging in, starting up applications, or issuing a PRINT command) can affect response times.

You can identify **hot** system files using a performance or monitoring tool (such as those listed in Section 1.5.2), and use the techniques in the following table to reduce hot file I/O activity on system disks:

| Potential Hot Files | Methods to Help |
|---|---|
| Page and swap files | When you run CLUSTER_CONFIG_LAN.COM or CLUSTER_CONFIG.COM to add computers to specify the sizes and locations of page and swap files, relocate the files as follows:<br>• Move page and swap files for computers off system disks.<br>• Set up page and swap files for satellites on the satellites' local disks, if such disks are available. |

| Potential Hot Files | Methods to Help |
|---|---|
| Move these high-activity files off the system disk:<br>• SYSUAF.DAT<br>• NETPROXY.DAT<br>• RIGHTSLIST.DAT<br>• ACCOUNTNG.DAT<br>• VMSMAIL_PROFILE.DATA<br>• QMAN$MASTER.DAT<br>• †VMS$OBJECTS.DAT<br>• Layered product and other application files | Use any of the following methods:<br>• Specify new locations for the files according to the instructions in Chapter 5.<br>• Use caching in the HSC subsystem or in RF or RZ disks to improve the effective system-disk throughput.<br>• Add a solid-state disk to your configuration. These devices have lower latencies and can handle a higher request rate than a regular magnetic disk. A solid-state disk can be used as a system disk or to hold system files.<br>• Use DECram software to create RAMdisks on MOP servers to hold copies of selected hot read-only files to improve boot times. A RAMdisk is an area of main memory within a system that is set aside to store data, but it is accessed as if it were a disk. |

†VAX specific

Moving these files from the system disk to a separate disk eliminates most of the write activity to the system disk. This raises the read/write ratio and, if you are using Volume Shadowing for OpenVMS, maximizes the performance of shadowing on the system disk.

### 9.5.3 Configuring Multiple System Disks

Depending on the number of computers to be included in a large cluster and the work being done, you must evaluate the tradeoffs involved in configuring a single system disk or multiple system disks.

While a single system disk is easier to manage, a large cluster often requires more system disk I/O capacity than a single system disk can provide. To achieve satisfactory performance, multiple system disks may be needed. However, you should recognize the increased system management efforts involved in maintaining multiple system disks.

Consider the following when determining the need for multiple system disks:

- Concurrent user activity

  In clusters with many satellites, the amount and type of user activity on those satellites influence system-disk load and, therefore, the number of satellites that can be supported by a single system disk. For example:

| IF... | THEN... | Comments |
|---|---|---|
| Many users are active or run multiple applications simultaneously | The load on the system disk can be significant; multiple system disks may be required. | Some OpenVMS Cluster systems may need to be configured on the assumption that all users are constantly active. Such working conditions may require a larger, more expensive OpenVMS Cluster system that handles peak loads without performance degradation. |

| IF... | THEN... | Comments |
|---|---|---|
| Few users are active simultaneously | A single system disk might support a large number of satellites. | For most configurations, the probability is low that most users are active simultaneously. A smaller and less expensive OpenVMS Cluster system can be configured for these typical working conditions but may suffer some performance degradation during peak load periods. |
| Most users run a single application for extended periods | A single system disk might support a large number of satellites if significant numbers of I/O requests can be directed to application data disks. | Because each workstation user in an OpenVMS Cluster system has a dedicated computer, a user who runs large compute-bound jobs on that dedicated computer does not significantly affect users of other computers in the OpenVMS Cluster system. For clustered workstations, the critical shared resource is a disk server. Thus, if a workstation user runs an I/O-intensive job, its effect on other workstations sharing the same disk server might be noticeable. |

- Concurrent booting activity

  One of the few times when all OpenVMS Cluster computers are simultaneously active is during a cluster reboot. All satellites are waiting to reload the operating system, and as soon as a boot server is available, they begin to boot in parallel. This booting activity places a significant I/O load on the boot server, system disk, and interconnect.

  **Note:** You can reduce overall cluster boot time by configuring multiple system disks and by distributing system roots for computers evenly across those disks. This technique has the advantage of increasing overall system disk I/O capacity, but it has the disadvantage of requiring additional system management effort. For example, installation of layered products or upgrades of the OpenVMS operating system must be repeated once for each system disk.

- System management

  Because system management work load increases as separate system disks are added and does so in direct proportion to the number of separate system disks that need to be maintained, you want to minimize the number of system disks added to provide the required level of performance.

Volume Shadowing for OpenVMS is an alternative to creating multiple system disks. Volume shadowing increases the read I/O capacity of a single system disk and minimizes the number of separate system disks that have to be maintained because installations or updates need only be applied once to a volume-shadowed system disk. For clusters with substantial system disk I/O requirements, you can use multiple system disks, each configured as a shadow set.

Cloning the system disk is a way to manage multiple system disks. To clone the system disk:

- Create a system disk (or shadow set) with roots for all OpenVMS Cluster nodes.
- Use this as a master copy, and perform all software upgrades on this system disk.
- Back up the master copy to the other disks to create the *cloned* system disks.
- Change the volume names so they are unique.

- If you have not moved system files off the system disk, you must have the SYLOGICALS.COM startup file point to system files on the master system disk.
- Before an upgrade, be sure to save any changes you need from the cloned disks since the last upgrade, such as MODPARAMS.DAT and AUTOGEN feedback data, accounting files for billing, and password history.

## 9.6 Conserving System Disk Space

The essential files for a satellite root take up very little space, so that more than 96 roots can easily fit on a single system disk. However, if you use separate dump files for each satellite node or put page and swap files for all the satellite nodes on the system disk, you quickly run out of disk space.

### 9.6.1 Techniques

To avoid running out of disk space, set up common dump files for all the satellites or for groups of satellite nodes. For debugging purposes, it is best to have separate dump files for each MOP and disk server. Also, you can use local disks on satellite nodes to hold page and swap files, instead of putting them on the system disk. In addition, move page and swap files for MOP and disk servers off the system disk.

**Reference:** See Section 10.8 to plan a strategy for managing dump files.

## 9.7 Adjusting System Parameters

As an OpenVMS Cluster system grows, certain data structures within OpenVMS need to grow in order to accommodate the large number of nodes. If growth is not possible (for example, because of a shortage of nonpaged pool) this will induce intermittent problems that are difficult to diagnose.

You should run AUTOGEN with FEEDBACK frequently as a cluster grows, so that settings for many parameters can be adjusted. Refer to Section 8.7 for more information about running AUTOGEN.

In addition to running AUTOGEN with FEEDBACK, you should check and manually adjust the following parameters:

- SCSBUFFCNT
- SCSRESPCNT
- CLUSTER_CREDITS

Prior to OpenVMS Version 7.2, customers were advised to also check the SCSCONNCNT parameter. However, beginning with OpenVMS Version 7.2, the SCSCONNCNT parameter is obsolete. SCS connections are now allocated and expanded only as needed, up to a limit of 65,000.

### 9.7.1 The SCSBUFFCNT Parameter (VAX Only)

**Note:** On Alpha systems, the SCS buffers are allocated as needed, and the SCSBUFFCNT parameter is reserved for OpenVMS use only.

**Description:** On VAX systems, the SCSBUFFCNT parameter controls the number of buffer descriptor table (BDT) entries that describe data buffers used in block data transfers between nodes.

**Symptoms of entry shortages:** A shortage of entries affects performance, most likely affecting nodes that perform MSCP serving.

**How to determine a shortage of BDT entries:** Use the SDA utility (or the Show Cluster utility) to identify systems that have waited for BDT entries.

```
SDA> READ SYS$SYSTEM:SCSDEF
%SDA-I-READSYM, reading symbol table  SYS$COMMON:[SYSEXE]SCSDEF.STB;1
SDA> EXAM @SCS$GL_BDT + CIBDT$L_QBDT_CNT
8046BB6C:  00000000   "...."
SDA>
```

**How to resolve shortages:** If the SDA EXAMINE command displays a nonzero value, BDT waits have occurred. If the number is nonzero and continues to increase during normal operations, increase the value of SCSBUFFCNT.

### 9.7.2 The SCSRESPCNT Parameter

**Description:** The SCSRESPCNT parameter controls the number of response descriptor table (RDT) entries available for system use. An RDT entry is required for every in-progress message exchange between two nodes.

**Symptoms of entry shortages:** A shortage of entries affects performance, since message transmissions must be delayed until a free entry is available.

**How to determine a shortage of RDT entries:** Use the SDA utility as follows to check each system for requests that waited because there were not enough free RDTs.

```
SDA> READ SYS$SYSTEM:SCSDEF
%SDA-I-READSYM, reading symbol table  SYS$COMMON:[SYSEXE]SCSDEF.STB;1
SDA> EXAM @SCS$GL_RDT + RDT$L_QRDT_CNT
8044DF74:  00000000   "...."
SDA>
```

**How to resolve shortages:** If the SDA EXAMINE command displays a nonzero value, RDT waits have occurred. If you find a count that tends to increase over time under normal operations, increase SCSRESPCNT.

### 9.7.3 The CLUSTER_CREDITS Parameter

**Description:** The CLUSTER_CREDITS parameter specifies the number of per-connection buffers a node allocates to receiving VMS$VAXcluster communications. This system parameter is not dynamic; that is, if you change the value, you must reboot the node on which you changed it.

**Default:** The default value is 10. The default value may be insufficient for a cluster that has very high locking rates.

**Symptoms of cluster credit problem:** A shortage of credits affects performance, since message transmissions are delayed until free credits are available. These are visible as credit waits in the SHOW CLUSTER display.

**How to determine whether credit waits exist:** Use the SHOW CLUSTER utility as follows:

1. Run SHOW CLUSTER/CONTINUOUS.
2. Type REMOVE SYSTEM/TYPE=HS.
3. Type ADD LOC_PROC, CR_WAIT.
4. Type SET CR_WAIT/WIDTH=10.

5. Check to see whether the number of CR_WAITS (credit waits) logged against the VMS$VAXcluster connection for any remote node is incrementing regularly. Ideally, credit waits should not occur. However, occasional waits under very heavy load conditions are acceptable.

**How to resolve incrementing credit waits:**

If the number of CR_WAITS is incrementing more than once per minute, perform the following steps:

1. Increase the CLUSTER_CREDITS parameter on the node against which they are being logged by five. The parameter should be modified on the remote node, not on the node which is running SHOW CLUSTER.
2. Reboot the node.

Note that it is not necessary for the CLUSTER_CREDITS parameter to be the same on every node.

## 9.8 Minimize Network Instability

Network instability also affects OpenVMS Cluster operations. Table 9–9 lists techniques to minimize typical network problems.

**Table 9–9 Techniques to Minimize Network Problems**

| Technique | Recommendation |
|---|---|
| Adjust the RECNXINTERVAL parameter. | The RECNXINTERVAL system parameter specifies the number of seconds the OpenVMS Cluster system waits when it loses contact with a node, before removing the node from the configuration. Many large OpenVMS Cluster configurations operate with the RECNXINTERVAL parameter set to 40 seconds (the default value is 20 seconds).<br>Raising the value of RECNXINTERVAL can result in longer perceived application pauses, especially when the node leaves the OpenVMS Cluster system abnormally. The pause is caused by the connection manager waiting for the number of seconds specified by RECNXINTERVAL. |
| Protect the network. | Treat the LAN as if it was a part of the OpenVMS Cluster system. For example, do not allow an environment in which a random user can disconnect a ThinWire segment to attach a new PC while 20 satellites hang. |
| Choose your hardware and configuration carefully. | Certain hardware is not suitable for use in a large OpenVMS Cluster system.<br>• Some network components can appear to work well with light loads, but are unable to operate properly under high traffic conditions. Improper operation can result in lost or corrupted packets that will require packet retransmissions. This reduces performance and can affect the stability of the OpenVMS Cluster configuration.<br>• Beware of bridges that cannot filter and forward at full line rates and repeaters that do not handle congested conditions well.<br>• Refer to *Guidelines for OpenVMS Cluster Configurations* to determine appropriate OpenVMS Cluster configurations and capabilities. |
| Use the LAVC$FAILURE_ ANALYSIS facility. | See Section D.5 for assistance in the isolation of network faults. |

## 9.9 DECnet Cluster Alias

You should define a cluster alias name for the OpenVMS Cluster to ensure that remote access will be successful when at least one OpenVMS Cluster member is available to process the client program's requests.

The cluster alias acts as a single network node identifier for an OpenVMS Cluster system. Computers in the cluster can use the alias for communications with other computers in a DECnet network. Note that it is possible for nodes running DECnet for OpenVMS to have a unique and separate cluster alias from nodes running DECnet–Plus. In addition, clusters running DECnet–Plus can have one cluster alias for VAX, one for Alpha, and another for both.

**Note:** A single cluster alias can include nodes running either DECnet for OpenVMS or DECnet–Plus, but not both. Also, an OpenVMS Cluster running both DECnet for OpenVMS and DECnet–Plus requires multiple system disks (one for each).

**Reference:** See Chapter 4 for more information about setting up and using a cluster alias in an OpenVMS Cluster system.

# 10

# Maintaining an OpenVMS Cluster System

Once your cluster is up and running, you can implement routine, site-specific maintenance operations—for example, backing up disks or adding user accounts, performing software upgrades and installations, running AUTOGEN with the feedback option on a regular basis, and monitoring the system for performance.

You should also maintain records of current configuration data, especially any changes to hardware or software components. If you are managing a cluster that includes satellite nodes, it is important to monitor LAN activity.

From time to time, conditions may occur that require the following special maintenance operations:

- Restoring cluster quorum after an unexpected computer failure
- Executing conditional shutdown operations
- Performing security functions in LAN and mixed-interconnect clusters

## 10.1 Backing Up Data and Files

As a part of the regular system management procedure, you should copy operating system files, application software files, and associated files to an alternate device using the OpenVMS Backup utility.

Some backup operations are the same in an OpenVMS Cluster as they are on a single OpenVMS system. For example, an incremental back up of a disk while it is in use, or the backup of a nonshared disk.

Backup tools for use in a cluster include those listed in Table 10–1.

**Table 10–1 Backup Methods**

| Tool | Usage |
|---|---|
| Online backup | Use from a running system to back up:<br>• The system's local disks<br>• Cluster-shareable disks other than system disks<br>• The system disk or disks<br>**Caution:** Files open for writing at the time of the backup procedure may not be backed up correctly. |

(continued on next page)

**Table 10–1 (Cont.) Backup Methods**

| Tool | Usage |
|---|---|
| Menu-driven or †standalone BACKUP | Use one of the following methods:<br>• If you have access to the OpenVMS Alpha or VAX distribution CD–ROM, back up your system using the menu system provided on that disc. This menu system, which is displayed automatically when you boot the CD–ROM, allows you to:<br>– Enter a DCL environment, from which you can perform backup and restore operations on the system disk (instead of using standalone BACKUP).<br>– Install or upgrade the operating system and layered products, using the POLYCENTER Software Installation utility.<br>**Reference:** For more detailed information about using the menu-driven procedure, see the *OpenVMS Upgrade and Installation Manual* and the *OpenVMS System Manager's Manual*.<br>• If you do not have access to the OpenVMS VAX distribution CD–ROM, you should use standalone BACKUP to back up and restore your system disk. Standalone BACKUP:<br>– Should be used with caution because it does not:<br>a. Participate in the cluster<br>b. Synchronize volume ownership or file I/O with other systems in the cluster<br>– Can boot from the system disk instead of the console media. Standalone BACKUP is built in the reserved root on any system disk.<br>**Reference:** For more information about standalone BACKUP, see the *OpenVMS System Manager's Manual*. |

†VAX specific

Plan to perform the backup process regularly, according to a schedule that is consistent with application and user needs. This may require creative scheduling so that you can coordinate backups with times when user and application system requirements are low.

**Reference:** See the *OpenVMS System Management Utilities Reference Manual: A–L* for complete information about the OpenVMS Backup utility.

## 10.2 Updating the OpenVMS Operating System

When updating the OpenVMS operating system, follow the steps in Table 10–2.

**Table 10–2 Upgrading the OpenVMS Operating System**

| Step | Action |
|---|---|
| **1** | Back up the system disk. |
| **2** | Perform the update procedure once for each system disk. |
| **3** | Install any mandatory updates. |

(continued on next page)

**Table 10–2 (Cont.) Upgrading the OpenVMS Operating System**

| Step | Action |
|---|---|
| **4** | Run AUTOGEN on each node that boots from that system disk. |
| **5** | Run the user environment test package (UETP) to test the installation. |
| **6** | Use the OpenVMS Backup utility to make a copy of the new system volume. |

**Reference:** See the appropriate OpenVMS upgrade and installation manual for complete instructions.

### 10.2.1 Rolling Upgrades

The OpenVMS operating system allows an OpenVMS Cluster system running on multiple system disks to continue to provide service while the system software is being upgraded. This process is called a **rolling upgrade** because each node is upgraded and rebooted in turn, until all the nodes have been upgraded.

If you must first migrate your system from running on one system disk to running on two or more system disks, follow these steps:

| Step | Action |
|---|---|
| **1** | Follow the procedures in Section 8.5 to create a duplicate disk. |
| **2** | Follow the instructions in Section 5.10 for information about coordinating system files. |

These sections help you add a system disk and prepare a common user environment on multiple system disks to make the shared system files such as the queue database, rightslists, proxies, mail, and other files available across the OpenVMS Cluster system.

## 10.3 LAN Network Failure Analysis

The OpenVMS operating system provides a sample program to help you analyze OpenVMS Cluster network failures on the LAN. You can edit and use the SYS$EXAMPLES:LAVC$FAILURE_ANALYSIS.MAR program to detect and isolate failed network components. Using the network failure analysis program can help reduce the time required to detect and isolate a failed network component, thereby providing a significant increase in cluster availability.

**Reference:** For a description of the network failure analysis program, refer to Appendix D.

## 10.4 Recording Configuration Data

To maintain an OpenVMS Cluster system effectively, you must keep accurate records about the current status of all hardware and software components and about any changes made to those components. Changes to cluster components can have a significant effect on the operation of the entire cluster. If a failure occurs, you may need to consult your records to aid problem diagnosis.

Maintaining current records for your configuration is necessary both for routine operations and for eventual troubleshooting activities.

### 10.4.1 Record Information

At a minimum, your configuration records should include the following information:

- A diagram of your physical cluster configuration. (Appendix D includes a discussion of keeping a LAN configuration diagram.)
- SCSNODE and SCSSYSTEMID parameter values for all computers.
- VOTES and EXPECTED_VOTES parameter values.
- DECnet names and addresses for all computers.
- Current values for cluster-related system parameters, especially ALLOCLASS and TAPE_ALLOCLASS values for HSC subsystems and computers.

  **Reference:** Cluster system parameters are described in Appendix A.
- Names and locations of default bootstrap command procedures for all computers connected with the CI.
- Names of cluster disk and tape devices.
- In LAN and mixed-interconnect clusters, LAN hardware addresses for satellites.
- Names of LAN adapters.
- Names of LAN segments or rings.
- Names of LAN bridges.
- Names of wiring concentrators or of DELNI or DEMPR adapters.
- Serial numbers of all hardware components.
- Changes to any hardware or software components (including site-specific command procedures), along with dates and times when changes were made.

### 10.4.2 Satellite Network Data

The first time you execute CLUSTER_CONFIG.COM to add a satellite, the procedure creates the file NETNODE_UPDATE.COM in the boot server's SYS$SPECIFIC:[SYSMGR] directory. (For a common-environment cluster, you must rename this file to the SYS$COMMON:[SYSMGR] directory, as described in Section 5.10.2.) This file, which is updated each time you add or remove a satellite or change its Ethernet or FDDI hardware address, contains all essential network configuration data for the satellite.

If an unexpected condition at your site causes configuration data to be lost, you can use NETNODE_UPDATE.COM to restore it. You can also read the file when you need to obtain data about individual satellites. Note that you may want to edit the file occasionally to remove obsolete entries.

Example 10–1 shows the contents of the file after satellites EUROPA and GANYMD have been added to the cluster.

**Example 10–1 Sample NETNODE_UPDATE.COM File**

```
$ RUN SYS$SYSTEM:NCP
    define node EUROPA address 2.21
    define node EUROPA hardware address 08-00-2B-03-51-75
    define node EUROPA load assist agent sys$share:niscs_laa.exe
    define node EUROPA load assist parameter $1$DJA11:<SYS10.>
    define node EUROPA tertiary loader sys$system:tertiary_vmb.exe
    define node GANYMD address 2.22
    define node GANYMD hardware address 08-00-2B-03-58-14
    define node GANYMD load assist agent sys$share:niscs_laa.exe
    define node GANYMD load assist parameter $1$DJA11:<SYS11.>
    define node GANYMD tertiary loader sys$system:tertiary_vmb.exe
```

**Reference:** See the DECnet–Plus documentation for equivalent NCL command information.

## 10.5 Cross-Architecture Satellite Booting

Cross-architecture satellite booting permits VAX boot nodes to provide boot service to Alpha satellites and Alpha boot nodes to provide boot service to VAX satellites. For some OpenVMS Cluster configurations, cross-architecture boot support can simplify day-to-day system operation and reduce the complexity of managing OpenVMS Cluster that include both VAX and Alpha systems.

**Note:** Compaq will continue to provide cross-architecture boot support while it is technically feasible. This support may be removed in future releases of the OpenVMS operating system.

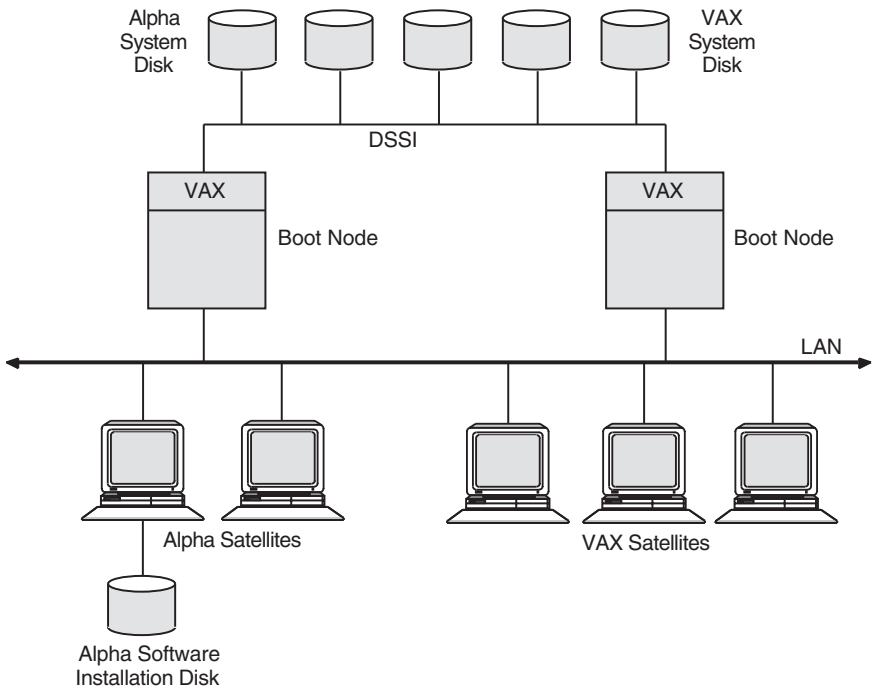
### 10.5.1 Sample Configurations

The sample configurations that follow show how you might configure an OpenVMS Cluster to include both Alpha and VAX boot nodes and satellite nodes. Note that each architecture must include a system disk that is used for installations and upgrades.

**Caution:** The OpenVMS operating system and layered product installations and upgrades cannot be performed across architectures. For example, OpenVMS Alpha software installations and upgrades must be performed using an Alpha system. When configuring OpenVMS Cluster systems that use the cross-architecture booting feature, configure at least one system of each architecture with a disk that can be used for installations and upgrades. In the configurations shown in Figure 10–1 and Figure 10–2, one of the workstations has been configured with a local disk for this purpose.

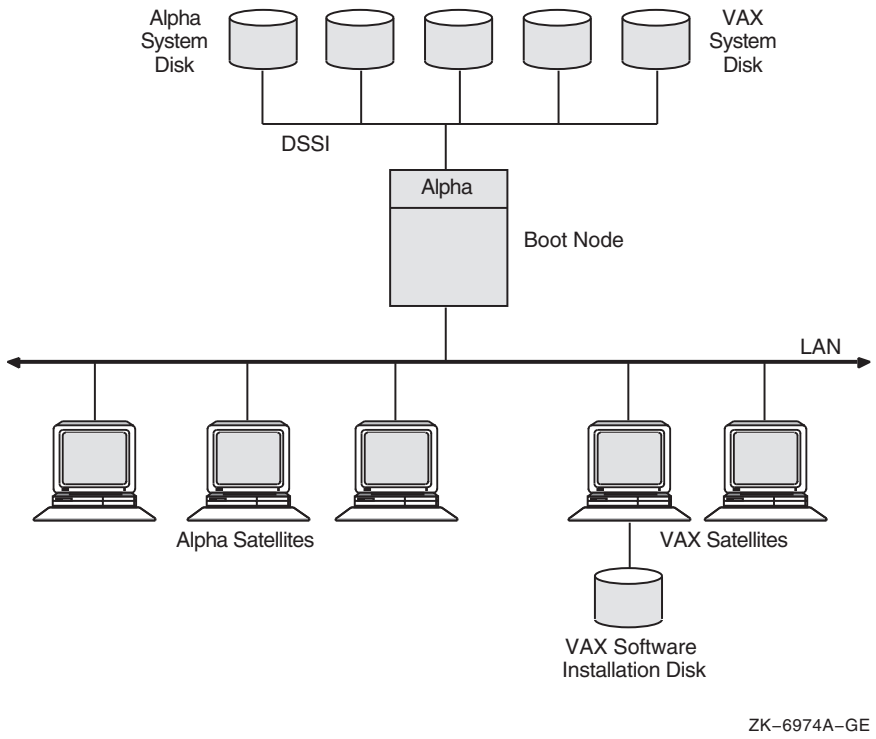
In Figure 10–1, several Alpha workstations have been added to an existing VAXcluster configuration that contains two VAX boot nodes based on the DSSI interconnect and several VAX workstations. For high availability, the Alpha system disk is located on the DSSI for access by multiple boot servers.

**Figure 10–1 VAX Nodes Boot Alpha Satellites**




In Figure 10–2, the configuration originally consisted of a VAX boot node and several VAX workstations. The VAX boot node has been replaced with a new, high-performance Alpha boot node. Some Alpha workstations have also been added. The original VAX workstations remain in the configuration and still require boot service. The new Alpha boot node can perform this service.

**Figure 10–2 Alpha and VAX Nodes Boot Alpha and VAX Satellites**




## 10.5.2 Usage Notes

Consider the following guidelines when using the cross-architecture booting feature:

- The OpenVMS software installation and upgrade procedures are architecture specific. The operating system must be installed and upgraded on a disk that is directly accessible from a system of the appropriate architecture. Configuring a boot server with a system disk of the opposite architecture involves three distinct system management procedures:
  - Installation of the operating system on a disk that is directly accessible from a system of the same architecture.
  - Moving the resulting system disk so that it is accessible by the target boot server. Depending on the specific configuration, this can be done using the Backup utility or by physically relocating the disk.
  - Setting up the boot server's network database to service satellite boot requests. Sample procedures for performing this step are included in Section 10.5.3.
- System disks can contain only a single version of the OpenVMS operating system and are architecture specific. For example, OpenVMS VAX Version 7.1 cannot coexist on a system disk with OpenVMS Alpha Version 7.1.
- The CLUSTER_CONFIG command procedure can be used only to manage cluster nodes of the same architecture as the node executing the procedure. For example, when run from an Alpha system, CLUSTER_CONFIG can manipulate only Alpha system disks and perform node management procedures for Alpha systems.

- No support is provided for cross-architecture installation of layered products.

### 10.5.3 Configuring DECnet

The following examples show how to configure DECnet databases to perform cross-architecture booting. Note that this feature is available for systems running DECnet for OpenVMS (Phase IV) only.

Customize the command procedures in Examples 10–2 and 10–3 according to the following instructions.

| Replace... | With... |
|---|---|
| *alpha_system_disk* or *vax_system_disk* | The appropriate disk name on the server |
| *label* | The appropriate label name for the disk on the server |
| *ccc-n* | The server circuit name |
| *alpha* or *vax* | The DECnet node name of the satellite |
| *xx.yyyy* | The DECnet area.address of the satellite |
| *aa-bb-cc-dd-ee-ff* | The hardware address of the LAN adapter on the satellite over which the satellite is to be loaded |
| *satellite_root* | The root on the system disk (for example, SYS10) of the satellite |

Example 10–2 shows how to set up a VAX system to serve a locally mounted Alpha system disk.

**Example 10–2 Defining an Alpha Satellite in a VAX Boot Node**

```
$! VAX system to load Alpha satellite
$!
$!  On the VAX system:
$!  -----------------
$!
$!  Mount the system disk for MOP server access.
$!
$ MOUNT /SYSTEM alpha_system_disk: label ALPHA$SYSD
$!
$!  Enable MOP service for this server.
$!
$ MCR NCP
NCP> DEFINE CIRCUIT ccc-n SERVICE ENABLED STATE ON
NCP> SET CIRCUIT ccc-n STATE OFF
NCP> SET CIRCUIT ccc-n ALL
NCP> EXIT
$!
$!  Configure MOP service for the ALPHA satellite.
$!
$ MCR NCP
NCP> DEFINE NODE alpha ADDRESS xx.yyyy
NCP> DEFINE NODE alpha HARDWARE ADDRESS aa-bb-cc-dd-ee-ff
NCP> DEFINE NODE alpha LOAD ASSIST AGENT SYS$SHARE:NISCS_LAA.EXE
NCP> DEFINE NODE alpha LOAD ASSIST PARAMETER ALPHA$SYSD:[satellite_root.]
NCP> DEFINE NODE alpha LOAD FILE APB.EXE
NCP> SET NODE alpha ALL
NCP> EXIT
```

Example 10–3 shows how to set up an Alpha system to serve a locally mounted VAX system disk.

**Example 10–3 Defining a VAX Satellite in an Alpha Boot Node**

```
$! Alpha system to load VAX satellite
$!
$!  On the Alpha system:
$!  --------------------
$!
$!  Mount the system disk for MOP server access.
$!
$ MOUNT /SYSTEM vax_system_disk: label VAX$SYSD
$!
$!  Enable MOP service for this server.
$!
$ MCR NCP
NCP> DEFINE CIRCUIT ccc-n SERVICE ENABLED STATE ON
NCP> SET CIRCUIT ccc-n STATE OFF
NCP> SET CIRCUIT ccc-n ALL
NCP> EXIT
$!
$!  Configure MOP service for the VAX satellite.
$!
$ MCR NCP
NCP> DEFINE NODE vax ADDRESS xx.yyyy
NCP> DEFINE NODE vax HARDWARE ADDRESS aa-bb-cc-dd-ee-ff
NCP> DEFINE NODE vax TERTIARY LOADER SYS$SYSTEM:TERTIARY_VMB.EXE
NCP> DEFINE NODE vax LOAD ASSIST AGENT SYS$SHARE:NISCS_LAA.EXE
NCP> DEFINE NODE vax LOAD ASSIST PARAMETER VAX$SYSD:[satellite_root.]
NCP> SET NODE vax ALL
NCP> EXIT
```

Then, to boot the satellite, perform these steps:

1. Execute the appropriate command procedure from a privileged account on the server
2. Boot the satellite over the adapter represented by the hardware address you entered into the command procedure earlier.

## 10.6 Controlling OPCOM Messages

When a satellite joins the cluster, the Operator Communications Manager (OPCOM) has the following default states:

- For all systems in an OpenVMS Cluster configuration except workstations:
  - OPA0: is enabled for all message classes.
  - The log file SYS$MANAGER:OPERATOR.LOG is opened for all classes.
- For workstations in an OpenVMS Cluster configuration, even though the OPCOM process is running:
  - OPA0: is not enabled.
  - No log file is opened.

### 10.6.1 Overriding OPCOM Defaults

Table 10–3 shows how to define the following system logical names in the command procedure SYS$MANAGER:SYLOGICALS.COM to override the OPCOM default states.

**Table 10–3 OPCOM System Logical Names**

| System Logical Name | Function |
|---|---|
| OPC$OPA0_ENABLE | If defined to be true, OPA0: is enabled as an operator console. If defined to be false, OPA0: is not enabled as an operator console. DCL considers any string beginning with T or Y or any odd integer to be true, all other values are false. |
| OPC$OPA0_CLASSES | Defines the operator classes to be enabled on OPA0:. The logical name can be a search list of the allowed classes, a list of classes, or a combination of the two. For example:<br>`$ DEFINE/SYSTEM OP$OPA0_CLASSES CENTRAL,DISKS,TAPE`<br>`$ DEFINE/SYSTEM OP$OPA0_CLASSES "CENTRAL,DISKS,TAPE"`<br>`$ DEFINE/SYSTEM OP$OPA0_CLASSES "CENTRAL,DISKS",TAPE`<br>You can define OPC$OPA0_CLASSES even if OPC$OPA0_ENABLE is not defined. In this case, the classes are used for any operator consoles that are enabled, but the default is used to determine whether to enable the operator console. |
| OPC$LOGFILE_ENABLE | If defined to be true, an operator log file is opened. If defined to be false, no log file is opened. |
| OPC$LOGFILE_CLASSES | Defines the operator classes to be enabled for the log file. The logical name can be a search list of the allowed classes, a comma-separated list, or a combination of the two. You can define this system logical even when the OPC$LOGFILE_ENABLE system logical is not defined. In this case, the classes are used for any log files that are open, but the default is used to determine whether to open the log file. |
| OPC$LOGFILE_NAME | Supplies information that is used in conjunction with the default name SYS$MANAGER:OPERATOR.LOG to define the name of the log file. If the log file is directed to a disk other than the system disk, you should include commands to mount that disk in the SYLOGICALS.COM command procedure. |

### 10.6.2 Example

The following example shows how to use the OPC$OPA0_CLASSES system logical to define the operator classes to be enabled. The following command prevents SECURITY class messages from being displayed on OPA0.

```
$ DEFINE/SYSTEM OPC$OPA0_CLASSES CENTRAL,PRINTER,TAPES,DISKS,DEVICES, -
_$ CARDS,NETWORK,CLUSTER,LICENSE,OPER1,OPER2,OPER3,OPER4,OPER5, -
_$ OPER6,OPER7,OPER8,OPER9,OPER10,OPER11,OPER12
```

In large clusters, state transitions (computers joining or leaving the cluster) generate many multiline OPCOM messages on a boot server's console device. You can avoid such messages by including the DCL command REPLY/DISABLE=CLUSTER in the appropriate site-specific startup command file or by entering the command interactively from the system manager's account.

## 10.7 Shutting Down a Cluster

The SHUTDOWN command of the SYSMAN utility provides five options for shutting down OpenVMS Cluster computers:

- NONE (the default)
- REMOVE_NODE

- CLUSTER_SHUTDOWN
- REBOOT_CHECK
- SAVE_FEEDBACK

These options are described in the following sections.

### 10.7.1 The NONE Option

If you select the default SHUTDOWN option NONE, the shutdown procedure performs the normal operations for shutting down a standalone computer. If you want to shut down a computer that you expect will rejoin the cluster shortly, you can specify the default option NONE. In that case, cluster quorum is not adjusted because the operating system assumes that the computer will soon rejoin the cluster.

In response to the "Shutdown options [NONE]:" prompt, you can specify the DISABLE_AUTOSTART=*n* option, where *n* is the number of minutes before autostart queues are disabled in the shutdown sequence. For more information about this option, see Section 7.13.

### 10.7.2 The REMOVE_NODE Option

If you want to shut down a computer that you expect will not rejoin the cluster for an extended period, use the REMOVE_NODE option. For example, a computer may be waiting for new hardware, or you may decide that you want to use a computer for standalone operation indefinitely.

When you use the REMOVE_NODE option, the active quorum in the remainder of the cluster is adjusted downward to reflect the fact that the removed computer's votes no longer contribute to the quorum value. The shutdown procedure readjusts the quorum by issuing the SET CLUSTER/EXPECTED_VOTES command, which is subject to the usual constraints described in Section 10.12.

**Note:** The system manager is still responsible for changing the EXPECTED_ VOTES system parameter on the remaining OpenVMS Cluster computers to reflect the new configuration.

### 10.7.3 The CLUSTER_SHUTDOWN Option

When you choose the CLUSTER_SHUTDOWN option, the computer completes all shut down activities up to the point where the computer would leave the cluster in a normal shutdown situation. At this point the computer waits until all other nodes in the cluster have reached the same point. When all nodes have completed their shutdown activities, the entire cluster dissolves in one synchronized operation. The advantage of this is that individual nodes do not complete shutdown independently, and thus do not trigger state transitions or potentially leave the cluster without quorum.

When performing a CLUSTER_SHUTDOWN you must specify this option on every OpenVMS Cluster computer. If any computer is not included, clusterwide shutdown cannot occur.

### 10.7.4 The REBOOT_CHECK Option

When you choose the REBOOT_CHECK option, the shutdown procedure checks for the existence of basic system files that are needed to reboot the computer successfully and notifies you if any files are missing. You should replace such files before proceeding. If all files are present, the following informational message appears:

```
%SHUTDOWN-I-CHECKOK, Basic reboot consistency check completed.
```

**Note:** You can use the REBOOT_CHECK option separately or in conjunction with either the REMOVE_NODE or the CLUSTER_SHUTDOWN option. If you choose REBOOT_CHECK with one of the other options, you must specify the options in the form of a comma-separated list.

### 10.7.5 The SAVE_FEEDBACK Option

Use the SAVE_FEEDBACK option to enable the AUTOGEN feedback operation.

**Note:** Select this option only when a computer has been running long enough to reflect your typical work load.

**Reference:** For detailed information about AUTOGEN feedback, see the *OpenVMS System Manager's Manual*.

## 10.8 Dump Files

Whether your OpenVMS Cluster system uses a single common system disk or multiple system disks, you should plan a strategy to manage dump files.

### 10.8.1 Controlling Size and Creation

Dump-file management is especially important for large clusters with a single system disk. For example, on a 256 MB OpenVMS Alpha computer, AUTOGEN creates a dump file in excess of 500,000 blocks.

In the event of a software-detected system failure, each computer normally writes the contents of memory to a full dump file on its system disk for analysis. By default, this full dump file is the size of physical memory plus a small number of pages. If system disk space is limited (as is probably the case if a single system disk is used for a large cluster), you may want to specify that no dump file be created for satellites or that AUTOGEN create a selective dump file. The selective dump file is typically 30% to 60% of the size of a full dump file.

You can control dump-file size and creation for each computer by specifying appropriate values for the AUTOGEN symbols DUMPSTYLE and DUMPFILE in the computer's MODPARAMS.DAT file. Specify dump files as shown in Table 10–4.

**Table 10–4 AUTOGEN Dump-File Symbols**

| Value Specified | Result |
|---|---|
| DUMPSTYLE = 0 | Full dump file created (default) |
| DUMPSTYLE = 1 | Selective dump file created |
| DUMPFILE = 0 | No dump file created |

**Caution:** Although you can configure computers without dump files, the lack of a dump file can make it difficult or impossible to determine the cause of a system failure.

For example, use the following commands to modify the system dump-file size on large-memory systems:

```
$ MCR SYSGEN
SYSGEN> USE CURRENT
SYSGEN> SET DUMPSTYLE 1
SYSGEN> CREATE SYS$SYSTEM:SYSDUMP.DMP/SIZE=70000
SYSGEN> WRITE CURRENT
SYSGEN> EXIT
$ @SHUTDOWN
```

The dump-file size of 70,000 blocks is sufficient to cover about 32 MB of memory. This size is usually large enough to encompass the information needed to analyze a system failure.

After the system reboots, you can purge SYSDUMP.DMP.

### 10.8.2 Sharing Dump Files

Another option for saving dump-file space is to share a single dump file among multiple computers. This technique makes it possible to analyze isolated computer failures. But dumps are lost if multiple computers fail at the same time or if a second computer fails before you can analyze the first failure. Because boot server failures have a greater impact on cluster operation than do failures of other computers you should configure full dump files on boot servers to help ensure speedy analysis of problems.

VAX systems cannot share dump files with Alpha computers and vice versa. However, you can share a single dump file among multiple Alpha computers and another single dump file among VAX computers. Follow these steps for each operating system:

| Step | Action |
|---|---|
| **1** | Decide whether to use full or selective dump files. |
| **2** | Determine the size of the largest dump file needed by any satellite. |
| **3** | Select a satellite whose memory configuration is the largest of any in the cluster and do the following:<br>1. Specify DUMPSTYLE = 0 (or DUMPSTYLE = 1) in that satellite's MODPARAMS.DAT file.<br>2. Remove any DUMPFILE symbol from the satellite's MODPARAMS.DAT file.<br>3. Run AUTOGEN on that satellite to create a dump file. |
| **4** | Rename the dump file to SYS$COMMON:[SYSEXE]SYSDUMP-COMMON.DMP or create a new dump file named SYSDUMP-COMMON.DMP in SYS$COMMON:[SYSEXE]. |

| Step | Action |
|---|---|
| **5** | For each satellite that is to share the dump file, do the following:<br>1. Create a file synonym entry for the dump file in the system-specific root. For example, to create a synonym for the satellite using root SYS1E, enter a command like the following:<br>`$ SET FILE SYS$COMMON:[SYSEXE]SYSDUMP-COMMON.DMP -`<br>`_$ /ENTER=SYS$SYSDEVICE:[SYS1E.SYSEXE]SYSDUMP.DMP`<br>2. Add the following lines to the satellite's MODPARAMS.DAT file:<br>`DUMPFILE = 0`<br>`DUMPSTYLE = 0 (or DUMPSTYLE = 1)` |
| **6** | Rename the old system-specific dump file on each system that has its own dump file:<br>`$ RENAME SYS$SYSDEVICE:[SYS`*n*`.SYSEXE]SYSDUMP.DMP .OLD`<br>The value of *n* in the command line is the root for each system (for example, SYS0 or SYS1). Rename the file so that the operating system software does not use it as the dump file when the system is rebooted. |
| **7** | Reboot each node so it can map to the new common dump file. The operating system software cannot use the new file for a crash dump until you reboot the system. |
| **8** | After you reboot, delete the SYSDUMP.OLD file in each system-specific root. Do not delete any file called SYSDUMP.DMP; instead, rename it, reboot, and then delete it as described in steps 6 and 7. |

## 10.9 Maintaining the Integrity of OpenVMS Cluster Membership

Because multiple LAN and mixed-interconnect clusters coexist on a single extended LAN, the operating system provides mechanisms to ensure the integrity of individual clusters and to prevent access to a cluster by an unauthorized computer.

The following mechanisms are designed to ensure the integrity of the cluster:

- A cluster authorization file (SYS$COMMON:[SYSEXE]CLUSTER_AUTHORIZE.DAT), which is initialized during installation of the operating system or during execution of the CLUSTER_CONFIG.COM CHANGE function. The file is maintained with the SYSMAN utility.
- Control of conversational bootstrap operations on satellites.

The purpose of the cluster group number and password is to prevent accidental access to the cluster by an unauthorized computer. Under normal conditions, the system manager specifies the cluster group number and password either during installation or when you run CLUSTER_CONFIG.COM (see Example 8–11) to convert a standalone computer to run in an OpenVMS Cluster system.

OpenVMS Cluster systems use these mechanisms to protect the integrity of the cluster in order to prevent problems that could otherwise occur under circumstances like the following:

- When setting up a new cluster, the system manager specifies a group number identical to that of an existing cluster on the same Ethernet.
- A satellite user with access to a local system disk tries to join a cluster by executing a conversational SYSBOOT operation at the satellite's console.

**Reference:** These mechanisms are discussed in Section 10.9.1 and Section 8.2.1, respectively.

### 10.9.1 Cluster Group Data

The cluster authorization file, SYS$COMMON:[SYSEXE]CLUSTER_AUTHORIZE.DAT, contains the cluster group number and (in scrambled form) the cluster password. The CLUSTER_AUTHORIZE.DAT file is accessible only to users with the SYSPRV privilege.

Under normal conditions, you need not alter records in the CLUSTER_AUTHORIZE.DAT file interactively. However, if you suspect a security breach, you may want to change the cluster password. In that case, you use the SYSMAN utility to make the change.

To change the cluster password, follow these instructions:

| Step | Action |
|---|---|
| **1** | Invoke the SYSMAN utility. |
| **2** | Log in as system manager on a boot server. |
| **3** | Enter the following command:<br>`$ RUN SYS$SYSTEM:SYSMAN`<br>`SYSMAN>` |
| **4** | At the SYSMAN> prompt, enter any of the CONFIGURATION commands in the following list.<br>• CONFIGURATION SET CLUSTER_AUTHORIZATION<br>Updates the cluster authorization file, CLUSTER_AUTHORIZE.DAT, in the directory SYS$COMMON:[SYSEXE]. (The SET command creates this file if it does not already exist.) You can include the following qualifiers on this command:<br>– /GROUP_NUMBER—Specifies a cluster group number. Group number must be in the range from 1 to 4095 or 61440 to 65535.<br>– /PASSWORD—Specifies a cluster password. Password may be from 1 to 31 characters in length and may include alphanumeric characters, dollar signs ( $ ), and underscores ( _ ).<br>• CONFIGURATION SHOW CLUSTER_AUTHORIZATION<br>Displays the cluster group number.<br>• HELP CONFIGURATION SET CLUSTER_AUTHORIZATION<br>Explains the command's functions. |
| **5** | If your configuration has multiple system disks, each disk must have a copy of CLUSTER_AUTHORIZE.DAT. You must run the SYSMAN utility to update all copies. |

**Caution:** If you change either the group number or the password, you must reboot the entire cluster. For instructions, see Section 8.6.

### 10.9.2 Example

Example 10–4 illustrates the use of the SYSMAN utility to change the cluster password.

**Example 10–4 Sample SYSMAN Session to Change the Cluster Password**

(continued on next page)

**Example 10–4 (Cont.) Sample SYSMAN Session to Change the Cluster Password**

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> SET ENVIRONMENT/CLUSTER
%SYSMAN-I-ENV, current command environment:
        Clusterwide on local cluster
        Username SYSTEM        will be used on nonlocal nodes
SYSMAN> SET PROFILE/PRIVILEGES=SYSPRV
SYSMAN> CONFIGURATION SET CLUSTER_AUTHORIZATION/PASSWORD=NEWPASSWORD
%SYSMAN-I-CAFOLDGROUP, existing group will not be changed
%SYSMAN-I-CAFREBOOT, cluster authorization file updated
 The entire cluster should be rebooted.
SYSMAN> EXIT
$
```

## 10.10 Adjusting Maximum Packet Size for LAN Configurations

You can adjust the maximum packet size for LAN configurations with the NISCS_MAX_PKTSZ system parameter.

### 10.10.1 System Parameter Settings for LANs

Starting with OpenVMS Version 7.3, the operating system (PEdriver) automatically detects the maximum packet size of all the virtual circuits to which the system is connected. If the maximum packet size of the system's interconnects is smaller than the default packet-size setting, PEdriver automatically reduces the default packet size.

For earlier versions of OpenVMS (VAX Version 6.0 to Version 7.2; Alpha Version 1.5 to Version 7.2-1), the NISCS_MAX_PKTSZ parameter should be set to 1498 for Ethernet clusters and to 4468 for FDDI clusters.

### 10.10.2 How to Use NISCS_MAX_PKTSZ

To obtain this parameter's current, default, minimum, and maximum values, issue the following command:

```
$ MC SYSGEN SHOW NISCS_MAX_PKTSZ
```

You can use the NISCS_MAX_PKTSZ parameter to reduce packet size, which in turn can reduce memory consumption. However, reducing packet size can also increase CPU utilization for block data transfers, because more packets will be required to transfer a given amount of data. Lock message packets are smaller than the minimum value, so the NISCS_MAX_PKTSZ setting will not affect locking performance.

You can also use NISCS_MAX_PKTSZ to force use of a common packet size on all LAN paths by bounding the packet size to that of the LAN path with the smallest packet size. Using a common packet size can avoid VC closure due to packet size reduction when failing down to a slower, smaller packet size network.

If a memory-constrained system, such as a workstation, has adapters to a network path with large-size packets, such as FDDI or Gigabit Ethernet with jumbo packets, then you may want to conserve memory by reducing the value of the NISCS_MAX_PKTSZ parameter.

### 10.10.3 Editing Parameter Files

If you decide to change the value of the NISCS_MAX_PKTSZ parameter, edit the SYS$SPECIFIC:[SYSEXE]MODPARAMS.DAT file to permit AUTOGEN to factor the changed packet size into its calculations.

## 10.11 Determining Process Quotas

On Alpha systems, process quota default values in SYSUAF.DAT are often higher than the SYSUAF.DAT defaults on VAX systems. How, then, do you choose values for processes that could run on Alpha systems or on VAX systems in an OpenVMS Cluster? Understanding how a process is assigned quotas when the process is created in a dual-architecture OpenVMS Cluster configuration will help you manage this task.

### 10.11.1 Quota Values

The quotas to be used by a new process are determined by the OpenVMS LOGINOUT software. LOGINOUT works the same on OpenVMS Alpha and OpenVMS VAX systems. When a user logs in and a process is started, LOGINOUT uses the *larger* of:

- The value of the quota defined in the process's SYSUAF.DAT record
- The *current* value of the corresponding PQL_M*quota* system parameter on the host node in the OpenVMS Cluster

**Example:** LOGINOUT compares the value of the account's ASTLM process limit (as defined in the common SYSUAF.DAT) with the value of the PQL_MASTLM system parameter on the host Alpha system or on the host VAX system in the OpenVMS Cluster.

### 10.11.2 PQL Parameters

The letter M in PQL_M means minimum. The PQL_M*quota* system parameters set a minumum value for the quotas. In the Current and Default columns of the following edited SYSMAN display, note how the current value of each PQL_M*quota* parameter exceeds its system-defined default value in most cases. Note that the following display is Alpha specific. A similar SYSMAN display on a VAX system would show "Pages" in the Unit column instead of "Pagelets".

```
SYSMAN> PARAMETER SHOW/PQL

%SYSMAN-I-USEACTNOD, a USE ACTIVE has been defaulted on node DASHER
Node DASHER:   Parameters in use: ACTIVE
Parameter Name         Current   Default   Minimum   Maximum Unit  Dynamic
--------------         -------   -------   -------   ------- ----  -------
PQL_MASTLM                 120         4        -1        -1 Ast         D
PQL_MBIOLM                 100         4        -1        -1 I/O         D
PQL_MBYTLM              100000      1024        -1        -1 Bytes       D
PQL_MCPULM                   0         0        -1        -1 10Ms        D
PQL_MDIOLM                 100         4        -1        -1 I/O         D
PQL_MFILLM                 100         2        -1        -1 Files       D
PQL_MPGFLQUOTA           65536      2048        -1        -1 Pagelets    D
PQL_MPRCLM                  10         0        -1        -1 Processes   D
PQL_MTQELM                   0         0        -1        -1 Timers      D
PQL_MWSDEFAULT            2000      2000        -1        -1 Pagelets
PQL_MWSQUOTA              4000      4000        -1        -1 Pagelets    D
PQL_MWSEXTENT             8192      4000        -1        -1 Pagelets    D
PQL_MENQLM                 300         4        -1        -1 Locks       D
PQL_MJTQUOTA                 0         0        -1        -1 Bytes       D
```

In this display, the values for many PQL_M*quota* parameters increased from the defaults to their current values. Typically, this happens over time when AUTOGEN feedback is run periodically on your system. The PQL_M*quota* values also can change, of course, when you modify the values in MODPARAMS.DAT or in SYSMAN. As you consider the use of a common SYSUAF.DAT in an OpenVMS Cluster with both VAX and Alpha computers, keep the dynamic nature of the PQL_M*quota* parameters in mind.

### 10.11.3 Examples

The following table summarizes common SYSUAF.DAT scenarios and probable results on VAX and Alpha computers in an OpenVMS Cluster system.

**Table 10–5 Common SYSUAF.DAT Scenarios and Probable Results**

| WHEN you set values at... | THEN a process that starts on... | Will result in... |
|---|---|---|
| Alpha level | An Alpha node | Execution with the values you deemed appropriate. |
| | A VAX node | LOGINOUT not using the system-specific PQL_M*quota* values defined on the VAX system because LOGINOUT finds higher values for each quota in the Alpha style SYSUAF.DAT. This could cause VAX processes in the OpenVMS Cluster to use inappropriately high resources. |
| VAX level | A VAX node | Execution with the values you deemed appropriate. |
| | An Alpha node | LOGINOUT ignoring the typically lower VAX level values in the SYSUAF and instead use the value of each quota's current PQL_M*quota* values on the Alpha system. Monitor the current values of PQL_M*quota* system parameters if you choose to try this approach. Increase as necessary the appropriate PQL_M*quota* values on the Alpha system in MODPARAMS.DAT. |

You might decide to experiment with the higher process-quota values that usually are associated with an OpenVMS Alpha system's SYSUAF.DAT as you determine values for a common SYSUAF.DAT in an OpenVMS Cluster environment. The higher Alpha-level process quotas might be appropriate for processes created on host VAX nodes in the OpenVMS Cluster if the VAX systems have large available memory resources.

You can determine the values that are appropriate for processes on your VAX and Alpha systems by experimentation and modification over time. Factors in your decisions about appropriate limit and quota values for each process will include the following:

- Amount of available memory
- CPU processing power
- Average work load of the applications
- Peak work loads of the applications

## 10.12 Restoring Cluster Quorum

During the life of an OpenVMS Cluster system, computers join and leave the cluster. For example, you may need to add more computers to the cluster to extend the cluster's processing capabilities, or a computer may shut down because of a hardware or fatal software error. The connection management software coordinates these cluster transitions and controls cluster operation.

When a computer shuts down, the remaining computers, with the help of the connection manager, reconfigure the cluster, excluding the computer that shut down. The cluster can survive the failure of the computer and continue process operations as long as the cluster votes total is greater than the cluster quorum value. If the cluster votes total falls below the cluster quorum value, the cluster suspends the execution of all processes.

### 10.12.1 Restoring Votes

For process execution to resume, the cluster votes total must be restored to a value greater than or equal to the cluster quorum value. Often, the required votes are added as computers join or rejoin the cluster. However, waiting for a computer to join the cluster and increasing the votes value is not always a simple or convenient remedy. An alternative solution, for example, might be to shut down and reboot all the computers with a reduce quorum value.

After the failure of a computer, you may want to run the Show Cluster utility and examine values for the VOTES, EXPECTED_VOTES, CL_VOTES, and CL_QUORUM fields. (See the *OpenVMS System Management Utilities Reference Manual* for a complete description of these fields.) The VOTES and EXPECTED_VOTES fields show the settings for each cluster member; the CL_VOTES and CL_QUORUM fields show the cluster votes total and the current cluster quorum value.

To examine these values, enter the following commands:

```
$ SHOW CLUSTER/CONTINUOUS

COMMAND> ADD CLUSTER
```

**Note:** If you want to enter SHOW CLUSTER commands interactively, you must specify the /CONTINUOUS qualifier as part of the SHOW CLUSTER command string. If you do not specify this qualifier, SHOW CLUSTER displays cluster status information returned by the DCL command SHOW CLUSTER and returns you to the DCL command level.

If the display from the Show Cluster utility shows the CL_VOTES value equal to the CL_QUORUM value, the cluster cannot survive the failure of any remaining voting member. If one of these computers shuts down, all process activity in the cluster stops.

### 10.12.2 Reducing Cluster Quorum Value

To prevent the disruption of cluster process activity, you can reduce the cluster quorum value as described in Table 10–6.

**Table 10–6 Reducing the Value of Cluster Quorum**

| Technique | Description |
|---|---|
| Use the DCL command SET CLUSTER/EXPECTED_VOTES to adjust the cluster quorum to a value you specify. | If you do not specify a value, the operating system calculates an appropriate value for you. You need to enter the command on only one computer to propagate the new value throughout the cluster. When you enter the command, the operating system reports the new value.<br>**Suggestion:** Normally, you use the SET CLUSTER/EXPECTED_VOTES command only after a computer has left the cluster for an extended period. (For more information about this command, see the *OpenVMS DCL Dictionary*.)<br>**Example:** For example, if you want to change expected votes to set the cluster quorum to 2, enter the following command:<br>`$ SET CLUSTER/EXPECTED_VOTES=3`<br>The resulting value for quorum is (3 + 2)/2 = 2.<br>**Note:** No matter what value you specify for the SET CLUSTER/EXPECTED_VOTES command, you cannot increase quorum to a value that is greater than the number of the votes present, nor can you reduce quorum to a value that is half or fewer of the votes present.<br>When a computer that previously was a cluster member is ready to rejoin, you must reset the EXPECTED_VOTES system parameter to its original value in MODPARAMS.DAT on all computers and then reconfigure the cluster according to the instructions in Section 8.6. You do not need to use the SET CLUSTER/EXPECTED_VOTES command to increase cluster quorum, because the quorum value is increased automatically when the computer rejoins the cluster. |
| Use the IPC Q command to recalculate the quorum. | Refer to the *OpenVMS System Manager's Manual, Volume 1: Essentials* for a description of the Q command. |
| Select one of the cluster-related shutdown options. | Refer to Section 10.7 for a description of the shutdown options. |

## 10.13 Cluster Performance

Sometimes performance issues involve monitoring and tuning applications and the system as a whole. Tuning involves collecting and reporting on system and network processes to improve performance. A number of tools can help you collect information about an active system and its applications.

### 10.13.1 Using the SHOW Commands

The following table briefly describes the SHOW commands available with the OpenVMS operating system. Use the SHOW DEVICE commands and qualifiers shown in the table.

| Command | Purpose |
|---|---|
| SHOW DEVICE/FULL | Shows the complete status of a device, including:<br>• Whether the disk is available to the cluster<br>• Whether the disk is MSCP served or dual ported<br>• The name and type (VAX or HSC) of the primary and secondary hosts<br>• Whether the disk is mounted on the system where you enter the command<br>• The systems in the cluster on which the disk is mounted |

| Command | Purpose |
|---|---|
| SHOW DEVICE/FILES | Displays a list of the names of all files open on a volume and their associated process name and process identifier (PID). The command:<br>• Lists files opened only on this node.<br>• Finds all open files on a disk. You can use either the SHOW DEVICE/FILES command or SYSMAN commands on each node that has the disk mounted. |
| SHOW DEVICE/SERVED | Displays information about disks served by the MSCP server on the node where you enter the command. Use the following qualifiers to customize the information:<br>• /HOST displays the names of processors that have devices online through the local MSCP server, and the number of devices.<br>• /RESOURCE displays the resources available to the MSCP server, total amount of nonpaged dynamic memory available for I/O buffers, and number of I/O request packets.<br>• /COUNT displays the number of each size and type of I/O operation the MSCP server has performed since it was started.<br>• /ALL displays all of the information listed for the SHOW DEVICE/SERVED command. |

The SHOW CLUSTER command displays a variety of information about the OpenVMS Cluster system. The display output provides a view of the cluster as seen from a single node, rather than a complete view of the cluster.

**Reference:** The *OpenVMS System Management Utilities Reference Manual* contains complete information about all the SHOW commands and the Show Cluster utility.

### 10.13.2 Using the Monitor Utility

The following table describes using the OpenVMS Monitor utility to locate disk I/O bottlenecks. I/O bottlenecks can cause the OpenVMS Cluster system to appear to hang.

| Step | Action |
|---|---|
| **1** | To determine which clusterwide disks may be problem disks:<br>1. Create a node-by-node summary of disk I/O using the MONITOR/NODE command<br>2. Adjust the "row sum" column for MSCP served disks as follows:<br>• I/O rate on serving node includes local requests and all requests from other nodes<br>• I/O rate on other nodes includes requests generated from that node<br>• Requests from remote nodes are counted twice in the row sum column<br>3. Note disks with the row sum more than 8 I/Os per second<br>4. Eliminate from the list of cluster problem disks the disks that are:<br>• Not shared<br>• Dedicated to an application<br>• In the process of being backed up |
| **2** | For each node, determine the impact of potential problem disks:<br>• If a disporportionate amount of a disk's I/O comes from a particular node, the problem is most likely specific to the node.<br>• If a disk's I/O is spread evenly over the cluster, the problem may be clusterwide overuse.<br>• If the average queue length for a disk on a given node is less than 0.2, then the disk is having little impact on the node. |
| **3** | For each problem disk, determine whether:<br>• Page and swap files from any node are on the disk.<br>• Commonly used programs or data files are on the disk (use the SHOW DEVICE/FILES command).<br>• Users with default directories on the disk are causing the problem. |

### 10.13.3 Using Compaq Availability Manager and DECamds

Compaq Availability Manager and DECamds are real-time monitoring, diagnostic, and correction tools used by system managers to improve the availability and throughput of a system. Availability Manager runs on OpenVMS Alpha or on a Windows node. DECamds runs on both OpenVMS VAX and OpenVMS Alpha and uses the DECwindows interface.

These products, which are included with the operating system, help system managers correct system resource utilization problems for CPU usage, low memory, lock contention, hung or runaway processes, I/O, disks, page files, and swap files.

Availability Manager enables you to monitor one or more OpenVMS nodes on an extended LAN from either an OpenVMS Alpha or a Windows node. Availability Manager collects system and process data from multiple OpenVMS nodes simultaneously. It analyzes the data and displays the output using a native Java GUI.

DECamds collects and analyzes data from multiple nodes (VAX and Alpha) simultaneously, directing all output to a centralized DECwindows display. DECamds helps you observe and troubleshoot availability problems, as follows:

- Alerts users to resource availability problems, suggests paths for further investigation, and recommends actions to improve availability.
- Centralizes management of remote nodes within an extended LAN.
- Allows real-time intervention, including adjustment of node and process parameters, even when remote nodes are hung.
- Adjusts to site-specific requirements through a wide range of customization options.

**Reference:** For more information about Availability Manager, see the *Availability Manager User's Guide* and the Availability Manager web site, which you can access from the Compaq OpenVMS site:

`http://www.openvms.compaq.com/openvms/`

For more information about DECamds, see the *DECamds User's Guide*.

### 10.13.4 Monitoring LAN Activity

It is important to monitor LAN activity on a regular basis. Using the SCA (Systems Communications Architecture) Control Program (SCACP), you can monitor LAN activity as well as set and show default ports, start and stop LAN devices, and assign priority values to channels.

**Reference:** For more information about SCACP, see the *OpenVMS System Management Utilities Reference Manual: A–L*.

Using NCP commands like the following, you can set up a convenient monitoring procedure to report activity for each 12-hour period. Note that DECnet event logging for event 0.2 (automatic line counters) must be enabled.

**Reference:** For detailed information on DECnet for OpenVMS event logging, refer to the *DECnet for OpenVMS Network Management Utilities* manual.

In these sample commands, BNA-0 is the line ID of the Ethernet line.

```
NCP> DEFINE LINE BNA-0 COUNTER TIMER 43200
NCP> SET LINE BNA-0 COUNTER TIMER 43200
```

At every timer interval (in this case, 12 hours), DECnet will create an event that sends counter data to the DECnet event log. If you experience a performance degradation in your cluster, check the event log for increases in counter values that exceed normal variations for your cluster. If all computers show the same increase, there may be a general problem with your Ethernet configuration. If, on the other hand, only one computer shows a deviation from usual values, there is probably a problem with that computer or with its Ethernet interface device.

The following layered products can be used in conjunction with one of Compaq's LAN bridges to monitor the LAN traffic levels: RBMS, DECelms, DECmcc, and LAN Traffic Monitor (LTM).

# A

# Cluster System Parameters

For systems to boot properly into a cluster, certain system parameters must be set on each cluster computer. Table A–1 lists system parameters used in cluster configurations.

## A.1 Values for Alpha and VAX Computers

Some system parameters for Alpha computers are in units of pagelets, whereas others are in pages. AUTOGEN determines the hardware page size and records it in the PARAMS.DAT file.

**Caution:** When reviewing AUTOGEN recommended values or when setting system parameters with SYSGEN, note carefully which units are required for each parameter.

Table A–1 describes system parameters that are specific to OpenVMS Cluster configurations that may require adjustment in certain configurations. Table A–2 describes OpenVMS Cluster specific system parameters that are reserved for OpenVMS use.

**Reference:** System parameters, including cluster and volume shadowing system parameters, are documented in the *OpenVMS System Management Utilities Reference Manual*.

**Table A–1 Adjustable Cluster System Parameters**

| Parameter | Description |
|---|---|
| ALLOCLASS | Specifies a numeric value from 0 to 255 to be assigned as the disk allocation class for the computer. The default value is 0. |
| CHECK_CLUSTER | Serves as a VAXCLUSTER parameter sanity check. When CHECK_ CLUSTER is set to 1, SYSBOOT outputs a warning message and forces a conversational boot if it detects the VAXCLUSTER parameter is set to 0. |
| CLUSTER_CREDITS | Specifies the number of per-connection buffers a node allocates to receiving VMS$VAXcluster communications.<br>If the SHOW CLUSTER command displays a high number of credit waits for the VMS$VAXcluster connection, you might consider increasing the value of CLUSTER_CREDITS on the other node. However, in large cluster configurations, setting this value unnecessarily high will consume a large quantity of nonpaged pool. Each receive buffer is at least SCSMAXMSG bytes in size but might be substantially larger depending on the underlying transport.<br>It is not required that all nodes in the cluster have the same value for CLUSTER_CREDITS. For small or memory-constrained systems, the default value of CLUSTER_CREDITS should be adequate. |

(continued on next page)

**Table A–1 (Cont.) Adjustable Cluster System Parameters**

| Parameter | Description |
|---|---|
| CWCREPRC_ENABLE | Controls whether an unprivileged user can create a process on another OpenVMS Cluster node. The default value of 1 allows an unprivileged user to create a detached process with the same UIC on another node. A value of 0 requires that a user have DETACH or CMKRNL privilege to create a process on another node. |
| DISK_QUORUM | The physical device name, in ASCII, of an optional quorum disk. ASCII spaces indicate that no quorum disk is being used. DISK_QUORUM must be defined on one or more cluster computers capable of having a direct (not MSCP served) connection to the disk. These computers are called **quorum disk watchers**. The remaining computers (computers with a blank value for DISK_QUORUM) recognize the name defined by the first watcher computer with which they communicate. |
| ‡DR_UNIT_BASE | Specifies the base value from which unit numbers for DR devices (StorageWorks RAID Array 200 Family logical RAID drives) are counted. DR_UNIT_BASE provides a way for unique RAID device numbers to be generated. DR devices are numbered starting with the value of DR_UNIT_BASE and then counting from there. For example, setting DR_UNIT_BASE to 10 will produce device names such as $1$DRA10, $1$DRA11, and so on. Setting DR_UNIT_BASE to appropriate, nonoverlapping values on all cluster members that share the same (nonzero) allocation class will ensure that no two RAID devices are given the same name. |
| EXPECTED_VOTES | Specifies a setting that is used to derive the initial quorum value. This setting is the sum of all VOTES held by potential cluster members.<br><br>By default, the value is 1. The connection manager sets a quorum value to a number that will prevent cluster partitioning (see Section 2.3). To calculate quorum, the system uses the following formula:<br><br>`estimated quorum = (EXPECTED_VOTES + 2)/2` |
| LOCKDIRWT | Lock manager directory system weight. Determines the portion of lock manager directory to be handled by this system. The default value is adequate for most systems. |
| †LRPSIZE | For VAX computers running VMS Version 5.5–2 and earlier, the LRPSIZE parameter specifies the size, in bytes, of the large request packets. The actual physical memory consumed by a large request packet is LRPSIZE plus overhead for buffer management. Normally, the default value is adequate. The value of LRPSIZE affects the transfer size used by VAX nodes on an FDDI ring.<br><br>FDDI supports transfers using large packets (up to 4468 bytes). PEDRIVER does not use large packets by default, but can take advantage of the larger packet sizes if you increase the LRPSIZE system parameter to 4474 or higher.<br><br>PEDRIVER uses the full FDDI packet size if the LRPSIZE is set to 4474 or higher. However, only FDDI nodes connected to the same ring use large packets. Nodes connected to an Ethernet segment restrict packet size to that of an Ethernet packet (1498 bytes). |

†VAX specific
‡Alpha specific

(continued on next page)

**Table A–1 (Cont.) Adjustable Cluster System Parameters**

| Parameter | Description |
|---|---|
| ‡MC_SERVICES_P0 (dynamic) | Controls whether other MEMORY CHANNEL nodes in the cluster continue to run if this node bugchecks or shuts down.<br>A value of 1 causes other nodes in the MEMORY CHANNEL cluster to fail with bugcheck code MC_FORCED_CRASH if this node bugchecks or shuts down.<br>The default value is 0. A setting of 1 is intended only for debugging purposes; the parameter should otherwise be left at its default state. |
| ‡MC_SERVICES_P2 (static) | Specifies whether to load the PMDRIVER (PMA0) MEMORY CHANNEL cluster port driver. PMDRIVER is a new driver that serves as the MEMORY CHANNEL cluster port driver. It works together with MCDRIVER (the MEMORY CHANNEL device driver and device interface) to provide MEMORY CHANNEL clustering. If PMDRIVER is not loaded, cluster connections will not be made over the MEMORY CHANNEL interconnect.<br>The default for MC_SERVICES_P2 is 1. This default value causes PMDRIVER to be loaded when you boot the system.<br>Compaq recommends that this value not be changed. This parameter value must be the same on all nodes connected by MEMORY CHANNEL. |
| ‡MC_SERVICES_P3 (dynamic) | Specifies the maximum number of tags supported. The maximum value is 2048 and the minimum value is 100.<br>The default value is 800. Compaq recommends that this value not be changed.<br>This parameter value must be the same on all nodes connected by MEMORY CHANNEL. |
| ‡MC_SERVICES_P4 (static) | Specifies the maximum number of regions supported. The maximum value is 4096 and the minimum value is 100.<br>The default value is 200. Compaq recommends that this value not be changed.<br>This parameter value must be the same on all nodes connected by MEMORY CHANNEL. |
| ‡MC_SERVICES_P6 (static) | Specifies MEMORY CHANNEL message size, the body of an entry in a free queue, or a work queue. The maximum value is 65536 and the minimum value is 544. The default value is 992, which is suitable in all cases except systems with highly constrained memory.<br>For such systems, you can reduce the memory consumption of MEMORY CHANNEL by slightly reducing the default value of 992. This value must always be equal to or greater than the result of the following calculation:<br>1. Select the larger of SCS_MAXMSG and SCS_MAXDG.<br>2. Round that value to the next quadword.<br>This parameter value must be the same on all nodes connected by MEMORY CHANNEL. |

‡Alpha specific

(continued on next page)

**Table A–1 (Cont.) Adjustable Cluster System Parameters**

| Parameter | Description |
|---|---|
| ‡MC_SERVICES_P7 (dynamic) | Specifies whether to suppress or display messages about cluster activities on this node. Can be set to a value of 0, 1, or 2. The meanings of these values are:<br>**Value** — **Meaning**<br>0 — Nonverbose mode—no informational messages will appear on the console or in the error log.<br>1 — Verbose mode—informational messages from both MCDRIVER and PMDRIVER will appear on the console and in the error log.<br>2 — Same as verbose mode plus PMDRIVER stalling and recovery messages.<br>The default value is 0. Compaq recommends that this value not be changed except for debugging MEMORY CHANNEL problems or adjusting the MC_SERVICES_P9 parameter. |
| ‡MC_SERVICES_P9 (static) | Specifies the number of initial entries in a single channel's free queue. The maximum value is 2048 and the minimum value is 10.<br>Note that MC_SERVICES_P9 is not a dynamic parameter; you must reboot the system after each change in order for the change to take effect.<br>The default value is 150. Compaq recommends that this value not be changed.<br>This parameter value must be the same on all nodes connected by MEMORY CHANNEL. |
| ‡MPDEV_AFB_INTVL (disks only) | Specifies the automatic failback interval in seconds. The automatic failback interval is the minimum number of seconds that must elapse before the system will attempt another failback from an MSCP path to a direct path on the same device.<br>MPDEV_POLLER must be set to ON to enable automatic failback. You can disable automatic failback without disabling the poller by setting MPDEV_AFB_INTVL to 0. The default is 300 seconds. |
| ‡MPDEV_D1 (disks only) | Reserved for use by the operating system. |
| ‡MPDEV_D2 (disks only) | Reserved for use by the operating system. |
| ‡MPDEV_D3 (disks only) | Reserved for use by the operating system. |
| ‡MPDEV_D4 (disks only) | Reserved for use by the operating system. |
| ‡MPDEV_ENABLE | Enables the formation of multipath sets when set to ON (1). When set to OFF (0), the formation of additional multipath sets and the addition of new paths to existing multipath sets is disabled. However, existing multipath sets remain in effect. The default is ON.<br>MPDEV_REMOTE and MPDEV_AFB_INTVL have no effect when MPDEV ENABLE is set to OFF. |
| ‡MPDEV_LCRETRIES (disks only) | Controls the number of times the system retries the direct paths to the controller that the logical unit is online to, before moving on to direct paths to the other controller, or to an MSCP served path to the device. The valid range for retries is 1 through 256. The default is 1. |

‡Alpha specific

(continued on next page)

**Table A–1 (Cont.) Adjustable Cluster System Parameters**

| Parameter | Description |
|---|---|
| ‡MPDEV_POLLER | Enables polling of the paths to multipath set members when set to ON (1). Polling allows early detection of errors on inactive paths. If a path becomes unavailable or returns to service, the system manager is notified with an OPCOM message. When set to OFF (0), multipath polling is disabled. The default is ON. Note that this parameter must be set to ON to use the automatic failback feature. |
| ‡MPDEV_REMOTE (disks only) | Enables MSCP served disks to become members of a multipath set when set to ON (1). When set to OFF (0), only local paths to a SCSI or Fibre Channel device are used in the formation of additional multipath sets. MPDEV_REMOTE is enabled by default. However, setting this parameter to OFF has no effect on existing multipath sets that have remote paths.<br>To use multipath failover to a served path, MPDEV_REMOTE must be enabled on all systems that have direct access to shared SCSI/Fibre Channel devices. The first release to provide this feature is OpenVMS Alpha Version 7.3-1. Therefore, all nodes on which MPDEV_REMOTE is enabled must be running OpenVMS Alpha Version 7.3-1 (or later).<br>If MPDEV_ENABLE is set to OFF (0), the setting of MPDEV_REMOTE has no effect because the addition of all new paths to multipath sets is disabled. The default is ON. |
| MSCP_BUFFER | This buffer area is the space used by the server to transfer data between client systems and local disks.<br>On VAX systems, MSCP_BUFFER specifies the number of pages to be allocated to the MSCP server's local buffer area.<br>On Alpha systems, MSCP_BUFFER specifies the number of pagelets to be allocated the MSCP server's local buffer area. |
| MSCP_CMD_TMO | Specifies the time in seconds that the OpenVMS MSCP server uses to detect MSCP command timeouts. The MSCP server must complete the command within a built-in time of approximately 40 seconds plus the value of the MSCP_CMD_TMO parameter.<br>An MSCP_CMD_TMO value of 0 is normally adequate. A value of 0 provides the same behavior as in previous releases of OpenVMS (which did not have an MSCP_CMD_TMO system parameter). A nonzero setting increases the amount of time before an MSCP command times out.<br>If command timeout errors are being logged on client nodes, setting the parameter to a nonzero value on OpenVMS servers reduces the number of errors logged. Increasing the value of this parameter reduces the numb client MSCP command timeouts and increases the time it takes to detect faulty devices.<br>If you need to decrease the number of command timeout errors, set an initial value of 60. If timeout errors continue to be logged, you can increase this value in increments of 20 seconds. |
| MSCP_CREDITS | Specifies the number of outstanding I/O requests that can be active from one client system. |
| MSCP_LOAD | Controls whether the MSCP server is loaded. Specify 1 to load the server, and use the default CPU load rating. A value greater than 1 loads the server and uses this value as a constant load rating. By default, the value is set to 0 and the server is not loaded. |

‡Alpha specific

(continued on next page)

**Table A–1 (Cont.) Adjustable Cluster System Parameters**

| Parameter | Description |
|---|---|
| MSCP_SERVE_ALL | Controls the serving of disks. The settings take effect when the system boots. You cannot change the settings when the system is running.<br>Starting with OpenVMS Version 7.2, the serving types are implemented as a bit mask. To specify the type of serving your system will perform, locate the type you want in the following table and specify its value. For some systems, you may want to specify two serving types, such as serving the system disk and serving locally attached disks. To specify such a combination, add the values of each type, and specify the sum.<br>In a mixed-version cluster that includes any systems running OpenVMS Version 7.1-*x* or earlier, serving all available disks is restricted to serving all disks except those whose allocation class does not match the system's node allocation class (pre-Version 7.2 meaning). To specify this type of serving, use the value 9 (which sets bit 0 and bit 3).<br>The following table describes the serving type controlled by each bit and its decimal value. |

| Bit and Value When Set | Description |
|---|---|
| Bit 0 (1) | Serve all available disks (locally attached and those connected to HS*x* and DSSI controllers). Disks with allocation classes that differ from the system's allocation class (set by the ALLOCLASS parameter) are also served if bit 3 is not set. |
| Bit 1 (2) | Serve locally attached (non-HS*x* and non-DSSI) disks. |
| Bit 2 (4) | Serve the system disk. This is the default setting. This setting is important when other nodes in the cluster rely on this system being able to serve its system disk. This setting prevents obscure contention problems that can occur when a system attempts to complete I/O to a remote system disk whose system has failed. |
| Bit 3 (8) | Restrict the serving specified by bit 0. All disks except those with allocation classes that differ from the system's allocation class (set by the ALLOCLASS parameter) are served.<br>This is pre-Version 7.2 behavior. If your cluster includes systems running Open 7.1-*x* or earlier, and you want to serve all available disks, you must specify 9, the result of setting this bit and bit 0. |

Although the serving types are now implemented as a bit mask, the values of 0, 1, and 2, specified by bit 0 and bit 1, retain their original meanings:

- 0 — Do not serve any disks (the default for earlier versions of OpenVMS).
- 1 — Serve all available disks.
- 2 — Serve only locally attached (non-HS*x* and non-DSSI) disks.

If the MSCP_LOAD system parameter is 0, MSCP_SERVE_ALL is ignored. For more information about this system parameter, see Section 6.3.1.

(continued on next page)

**Table A–1 (Cont.) Adjustable Cluster System Parameters**

| Parameter | Description |
|---|---|
| NISCS_CONV_BOOT | During booting as an OpenVMS Cluster satellite, specifies whether conversational bootstraps are enabled on the computer. The default value of 0 specifies that conversational bootstraps are disabled. A value of 1 enables conversational bootstraps. |
| NISCS_LAN_OVRHD | Starting with OpenVMS Version 7.3, this parameter is obsolete. This parameter was formerly provided to reserve space in a LAN packet for encryption fields applied by external encryption devices. PEDRIVER now automatically determines the maximum packet size a LAN path can deliver, including any packet-size reductions required by external encryption devices. |
| NISCS_LOAD_PEA0 | Specifies whether the port driver (PEDRIVER) is to be loaded to enable cluster communications over the local area network (LAN). The default value of 0 specifies that the driver is not loaded. A value of 1 specifies that that driver is loaded.<br>**Caution:** If the NISCS_LOAD_PEA0 parameter is set to 1, the VAXCLUSTER system parameter must be set to 2. This ensures coordinated access to shared resources in the OpenVMS Cluster and prevents accidental data corruption. |

(continued on next page)

**Table A–1 (Cont.) Adjustable Cluster System Parameters**

| Parameter | Description |
|---|---|
| NISCS_MAX_PKTSZ | Specifies an upper limit, in bytes, on the size of the user data area in the largest packet sent by NISCA on any local area network (LAN).<br>The NISCS_MAX_PKTSZ parameter allows the system manager to change the packet size used for cluster communications on network communication paths. PEDRIVER automatically allocates memory to support the largest packet size that is usable by any virtual circuit connected to the system up to the limit set by this parameter.<br>Its default values are different for OpenVMS Alpha and OpenVMS VAX.<br>• On Alpha, the default value is the largest packet size currently supported by OpenVMS, in order to optimize performance.<br>• On VAX, the default value is the Ethernet packet size.<br>PEDRIVER uses NISCS_MAX_PKTSZ to compute the maximum amount of data to transmit in any LAN packet as follows:<br>LAN packet size <= (LAN header (padded Ethernet format) + NISCS_ MAX_PKTSZ + NISCS checksum (only if data checking is enabled) + LAN CRC or FCS)<br>The actual packet size automatically used by PEDRIVER might be smaller than the NISCS_MAX_PKTSZ limit for either of the following reasons:<br>• On a per-LAN-path basis, if PEdriver determines that the LAN path between two nodes, including the local and remote LAN adapters and intervening LAN equipment, can convey only a lesser size.<br>Only nodes with large-packet LAN adapters connected end-to-end by large-packet LAN equipment can use large packets. Nodes connected to large-packet LANs but having an end-to-end path that involves an Ethernet segment restrict packet size to that of an Ethernet packet (1498 bytes).<br>• For performance reasons, PEDRIVER might further limit the upper bound on packet size so that the packets can be allocated from a lookaside list in the nonpaged pool.<br>The actual memory allocation includes the required data structure overhead used by PEDRIVER and the LAN drivers, in addition to the actual LAN packet size.<br>The following table shows the minimum NISCS_MAX_PKTSZ value required to use the maximum packet size supported by LAN types.<br>**Type of LAN** — **Minimum Value for NISCS_MAX_PKTSZ**<br>Ethernet — 1498<br>FDDI — 4468<br>Gigabit Ethernet — 7532<br>ATM — 7606 |

(continued on next page)

**Table A–1 (Cont.) Adjustable Cluster System Parameters**

| Parameter | Description |
|---|---|
| NISCS_PORT_SERV | NISCS_PORT_SERV provides flag bits for PEDRIVER port services. Setting bits 0 and 1 (decimal value 3) enables data checking. The remaining bits are reserved for future use. Starting with OpenVMS Version 7.3-1, you can use the SCACP command SET VC/CHECKSUMMING to specify data checking on the VCs to certain nodes. You can do this on a running system. (Refer to the SCACP documentation in the *OpenVMS System Management Utilities Reference Manual* for more information.)<br>On the other hand, changing the setting of NISCS_PORT_SERV requires a reboot. Furthermore, this parameter applies to all virtual circuits between the node on which it is set and other nodes in the cluster.<br>NISCS_PORT_SERV has the AUTOGEN attribute. |
| PASTDGBUF | Specifies the number of datagram receive buffers to queue initially for the cluster port driver's configuration poller. The initial value is expanded during system operation, if needed.<br>MEMORY CHANNEL devices ignore this parameter. |
| QDSKINTERVAL | Specifies, in seconds, the disk quorum polling interval. The maximum is 32767, the minimum is 1, and the default is 3. Lower values trade increased overhead cost for greater responsiveness.<br>This parameter should be set to the same value on each cluster computer. |
| QDSKVOTES | Specifies the number of votes contributed to the cluster votes total by a quorum disk. The maximum is 127, the minimum is 0, and the default is 1. This parameter is used only when DISK_QUORUM is defined. |
| RECNXINTERVAL | Specifies, in seconds, the interval during which the connection manager attempts to reconnect a broken connection to another computer. If a new connection cannot be established during this period, the connection is declared irrevocably broken, and either this computer or the other must leave the cluster. This parameter trades faster response to certain types of system failures for the ability to survive transient faults of increasing duration.<br>This parameter should be set to the same value on each cluster computer. This parameter also affects the tolerance of the OpenVMS Cluster system for LAN bridge failures (see Section 3.4.7). |
| SCSBUFFCNT | †On VAX systems, SCSBUFFCNT is the number of buffer descriptors configured for all SCS devices. If no SCS device is configured on your system, this parameter is ignored. Generally, each data transfer needs a buffer descriptor: thus, the number of buffer descriptors limit the number of possible simultaneous I/Os. Various performance monitors report when a system is out of buffer descriptors for a given work load, indicating that a larger value for SCSBUFFCNT is worth considering.<br>**Note:** AUTOGEN provides feedback for this parameter on VAX systems only.<br>‡On Alpha systems, the SCS buffers are allocated as needed, and SCSBUFFCNT is reserved for OpenVMS use, only. |

†VAX specific
‡Alpha specific

(continued on next page)

**Table A–1 (Cont.) Adjustable Cluster System Parameters**

| Parameter | Description |
|---|---|
| SCSCONNCNT | The initial number of SCS connections that are configured for use by all system applications, including the one used by Directory Service Listen. The initial number will be expanded by the system if needed.<br>If no SCS ports are configured on your system, this parameter is ignored. The default value is adequate for all SCS hardware combinations.<br>**Note:** AUTOGEN provides feedback for this parameter on VAX systems only. |
| SCSNODE[1] | Specifies the name of the computer. This parameter is not dynamic.<br>Specify SCSNODE as a string of up to six characters. Enclose the string in quotation marks.<br>If the computer is in an OpenVMS Cluster, specify a value that is unique within the cluster. Do not specify the null string.<br>If the computer is running DECnet for OpenVMS, the value must be the same as the DECnet node name. |
| SCSRESPCNT | SCSRESPCNT is the total number of response descriptor table entries (RDTEs) configured for use by all system applications.<br>If no SCS or DSA port is configured on your system, this parameter is ignored. |
| SCSSYSTEMID[1] | Specifies a number that identifies the computer. This parameter is not dynamic. SCSSYSTEMID is the low-order 32 bits of the 48-bit system identification number.<br>If the computer is in an OpenVMS Cluster, specify a value that is unique within the cluster.<br>If the computer is running DECnet for OpenVMS, calculate the value from the DECnet address using the following formula:<br>`SCSSYSTEMID = (`*`DECnet-area-number`*` * 1024)`<br>`+ `*`DECnet-node-number`*<br>**Example:** If the DECnet address is 2.211, calculate the value as follows:<br>`SCSSYSTEMID = (2 * 1024) + 211 = 2259` |
| SCSSYSTEMIDH | Specifies the high-order 16 bits of the 48-bit system identification number. This parameter must be set to 0. It is reserved by OpenVMS for future use. |
| TAPE_ALLOCLASS | Specifies a numeric value from 0 to 255 to be assigned as the tape allocation class for tape devices connected to the computer. The default value is 0. |
| TIMVCFAIL | Specifies the time required for a virtual circuit failure to be detected. Compaq recommends that you use the default value. Compaq further recommends that you decrease this value only in OpenVMS Cluster systems of three or fewer CPUs, use the same value on each computer in the cluster, and use dedicated LAN segments for cluster I/O. |
| TMSCP_LOAD | Controls whether the TMSCP server is loaded. Specify a value of 1 to load the server and set all available TMSCP tapes served. By default, the value is set to 0, and the server is not loaded. |

[1]Once a computer has been recognized by another computer in the cluster, you cannot change the SCSSYSTEMID or SCSNODE parameter without either changing both or rebooting the entire cluster.

(continued on next page)

**Table A–1 (Cont.) Adjustable Cluster System Parameters**

| Parameter | Description |
|---|---|
| TMSCP_SERVE_ALL | Controls the serving of tapes. The settings take effect when the system boots. You cannot change the settings when the system is running.<br>Starting with OpenVMS Version 7.2, the serving types are implemented as a bit mask. To specify the type of serving your system will perform, locate the type you want in the following table and specify its value. For some systems, you may want to specify two serving types, such as serving all tapes except those whose allocation class does not match. To specify such a combination, add the values of each type, and specify the sum.<br>In a mixed-version cluster that includes any systems running OpenVMS Version 7.1-*x* or earlier, serving all available tapes is restricted to serving all tapes except those whose allocation class does not match the system's allocation class (pre-Version 7.2 meaning). To specify this type of serving, use the value 9, which sets bit 0 and bit 3.<br>The following table describes the serving type controlled by each bit and its decimal value. |

| Bit | Value When Set | Description |
|---|---|---|
| Bit 0 | 1 | Serve all available tapes (locally attached and those connected to HS*x* and DSSI controllers). Tapes with allocation classes that differ from the system's allocation class (set by the ALLOCLASS parameter) are also served if bit 3 is not set. |
| Bit 1 | 2 | Serve locally attached (non-HS*x* and non-DSSI) tapes. |
| Bit 2 | n/a | Reserved. |
| Bit 3 | 8 | Restrict the serving specified by bit 0. All tapes except those with allocation classes that differ from the system's allocation class (set by the ALLOCLASS parameter) are served.<br>This is pre-Version 7.2 behavior. If your cluster includes systems running OpenVMS Version 7.1-*x* or earlier, and you want to serve all available tapes, you must specify 9, the result of setting this bit and bit 0. |

Although the serving types are now implemented as a bit mask, the values of 0, 1, and 2, specified by bit 0 and bit 1, retain their original meanings:

- 0 — Do not serve any disks (the default for earlier versions of OpenVMS).
- 1 — Serve all available disks.
- 2 — Serve only locally attached (non-HS*x* and non-DSSI) disks.

If the TMSCP_LOAD system parameter is 0, TMSCP_SERVE_ALL is ignored.

(continued on next page)

**Table A–1 (Cont.) Adjustable Cluster System Parameters**

| Parameter | Description |
|---|---|
| VAXCLUSTER | Controls whether the computer should join or form a cluster. This parameter accepts the following three values:<br>• 0 — Specifies that the computer will not participate in a cluster.<br>• 1 — Specifies that the computer should participate in a cluster if hardware supporting SCS (CI or DSSI) is present or if NISCS_LOAD_PEA0 is set to 1, indicating that cluster communications is enabled over the local area network (LAN).<br>• 2 — Specifies that the computer should participate in a cluster.<br>You should always set this parameter to 2 on computers intended to run in a cluster, to 0 on computers that boot from a UDA disk controller and are not intended to be part of a cluster, and to 1 (the default) otherwise.<br>**Caution:** If the NISCS_LOAD_PEA0 system parameter is set to 1, the VAXCLUSTER parameter must be set to 2. This ensures coordinated access to shared resources in the OpenVMS Cluster system and prevents accidental data corruption. Data corruption may occur on shared resources if the NISCS_LOAD_PEA0 parameter is set to 1 and the VAXCLUSTER parameter is set to 0. |
| VOTES | Specifies the number of votes toward a quorum to be contributed by the computer. The default is 1. |

Table A–2 lists system parameters that should not require adjustment at any time. These parameters are provided for use in system debugging. Compaq recommends that you do not change these parameters unless you are advised to do so by your Compaq support representative. Incorrect adjustment of these parameters can result in cluster failures.

**Table A–2 Cluster System Parameters Reserved for OpenVMS Use Only**

| Parameter | Description |
|---|---|
| ‡MC_SERVICES_P1 (dynamic) | The value of this parameter must be the same on all nodes connected by MEMORY CHANNEL. |
| ‡MC_SERVICES_P5 (dynamic) | This parameter must remain at the default value of 8000000. This parameter value must be the same on all nodes connected by MEMORY CHANNEL. |
| ‡MC_SERVICES_P8 (static) | This parameter must remain at the default value of 0. This parameter value must be the same on all nodes connected by MEMORY CHANNEL. |
| ‡MPDEV_D1 | A multipath system parameter. |

‡Alpha specific

(continued on next page)

**Table A–2 (Cont.) Cluster System Parameters Reserved for OpenVMS Use Only**

| Parameter | Description |
|---|---|
| PAMAXPORT | PAMAXPORT specifies the maximum port number to be polled on each CI and DSSI. The CI and DSSI port drivers poll to discover newly initialized ports or the absence or failure of previously responding remote ports.<br>A system will not detect the existence of ports whose port numbers are higher than this parameter's value. Thus, this parameter should be set to a value that is greater than or equal to the highest port number being used on any CI or DSSI connected to the system.<br>You can decrease this parameter to reduce polling activity if the hardware configuration has fewer than 16 ports. For example, if the CI or DSSI with the largest configuration has a total of five ports assigned to port numbers 0 through 4, you could set PAMAXPORT to 4.<br>If no CI or DSSI devices are configured on your system, this parameter is ignored.<br>The default for this parameter is 15 (poll for all possible ports 0 through 15). Compaq recommends that you set this parameter to the same value on each cluster computer. |
| PANOPOLL | Disables CI and DSSI polling for ports if set to 1. (The default is 0.) When PANOPOLL is set, a computer will not promptly discover that another computer has shut down or powered down and will not discover a new computer that has booted. This parameter is useful when you want to bring up a computer detached from the rest of the cluster for debugging purposes.<br>PANOPOLL is functionally equivalent to uncabling the system from the DSSI or star coupler. This parameter does not affect OpenVMS Cluster communications over the LAN.<br>The default value of 0 is the normal setting and is required if you are booting from an HSC controller or if your system is joining an OpenVMS Cluster. This parameter is ignored if there are no CI or DSSI devices configured on your system. |
| PANUMPOLL | Establishes the number of CI and DSSI ports to be polled during each polling interval. The normal setting for PANUMPOLL is 16.<br>On older systems with less powerful CPUs, the parameter may be useful in applications sensitive to the amount of contiguous time that the system spends at IPL 8. Reducing PANUMPOLL reduces the amount of time spent at IPL 8 during each polling interval while increasing the number of polling intervals needed to discover new or failed ports.<br>If no CI or DSSI devices are configured on your system, this parameter is ignored. |
| PAPOLLINTERVAL | Specifies, in seconds, the polling interval the CI port driver uses to poll for a newly booted computer, a broken port-to-port virtual circuit, or a failed remote computer.<br>This parameter trades polling overhead against quick response to virtual circuit failures. This parameter should be set to the same value on each cluster computer. |
| PAPOOLINTERVAL | Specifies, in seconds, the interval at which the port driver checks available nonpaged pool after a pool allocation failure.<br>This parameter trades faster response to pool allocation failures for increased system overhead.<br>If CI or DSSI devices are not configured on your system, this parameter is ignored. |

(continued on next page)

**Table A–2 (Cont.) Cluster System Parameters Reserved for OpenVMS Use Only**

| Parameter | Description |
|---|---|
| PASANITY | PASANITY controls whether the CI and DSSI port sanity timers are enabled to permit remote systems to detect a system that has been hung at IPL 8 or higher for 100 seconds. It also controls whether virtual circuit checking gets enabled on the local system. The TIMVCFAIL parameter controls the time (1–99 seconds).<br>PASANITY is normally set to 1 and should be set to 0 only if you are debugging with XDELTA or planning to halt the CPU for periods of 100 seconds or more.<br>PASANITY is only semidynamic. A new value of PASANITY takes effect on the next CI or DSSI port reinitialization.<br>If CI or DSSI devices are not configured on your system, this parameter is ignored. |
| PASTDGBUF | The number of datagram receive buffers to queue initially for each CI or DSSI port driver's configuration poller; the initial value is expanded during system operation, if needed.<br>If no CI or DSSI devices are configured on your system, this parameter is ignored. |
| PASTIMOUT | The basic interval at which the CI port driver wakes up to perform time-based bookkeeping operations. It is also the period after which a timeout will be declared if no response to a start handshake datagram has been received.<br>If no CI or DSSI device is configured on your system, this parameter is ignored. |
| PRCPOLINTERVAL | Specifies, in seconds, the polling interval used to look for SCS applications, such as the connection manager and MSCP disks, on other computers. Each computer is polled, at most, once each interval.<br>This parameter trades polling overhead against quick recognition of new computers or servers as they appear. |
| SCSMAXMSG | The maximum number of bytes of system application data in one sequenced message. The amount of physical memory consumed by one message is SCSMAXMSG plus the overhead for buffer management.<br>If an SCS port is not configured on your system, this parameter is ignored. |
| SCSMAXDG | Specifies the maximum number of bytes of application data in one datagram.<br>If an SCS port is not configured on your system, this parameter is ignored. |
| SCSFLOWCUSH | Specifies the lower limit for receive buffers at which point SCS starts to notify the remote SCS of new receive buffers. For each connection, SCS tracks the number of receive buffers available. SCS communicates this number to the SCS at the remote end of the connection. However, SCS does not need to do this for each new receive buffer added. Instead, SCS notifies the remote SCS of new receive buffers if the number of receive buffers falls as low as the SCSFLOWCUSH value.<br>If an SCS port is not configured on your system, this parameter is ignored. |

# B
# Building Common Files

This appendix provides guidelines for building a common user authorization file (UAF) from computer-specific files. It also describes merging RIGHTSLIST.DAT files.

For more detailed information about how to set up a computer-specific authorization file, see the descriptions in the *OpenVMS Guide to System Security*.

## B.1 Building a Common SYSUAF.DAT File

To build a common SYSUAF.DAT file, follow the steps in Table B–1.

**Table B–1 Building a Common SYSUAF.DAT File**

| Step | Action |
|---|---|
| **1** | Print a listing of SYSUAF.DAT on each computer. To print this listing, invoke AUTHORIZE and specify the AUTHORIZE command LIST as follows:<br><br>`$ SET DEF SYS$SYSTEM`<br>`$ RUN AUTHORIZE`<br>`UAF> LIST/FULL [*,*]` |

(continued on next page)

**Table B–1 (Cont.) Building a Common SYSUAF.DAT File**

| Step | Action |
|---|---|
| **2** | Use the listings to compare the accounts from each computer. On the listings, mark any necessary changes. For example:<br>• Delete any accounts that you no longer need.<br>• Make sure that UICs are set appropriately:<br>– User UICs<br>Check each user account in the cluster to see whether it should have a unique user identification code (UIC). For example, OpenVMS Cluster member VENUS may have a user account JONES that has the same UIC as user account SMITH on computer MARS. When computers VENUS and MARS are joined to form a cluster, accounts JONES and SMITH will exist in the cluster environment with the same UIC. If the UICs of these accounts are not differentiated, each user will have the same access rights to various objects in the cluster. In this case, you should assign each account a unique UIC.<br>– Group UICs<br>Make sure that accounts that perform the same type of work have the same group UIC. Accounts in a single-computer environment probably follow this convention. However, there may be groups of users on each computer that will perform the same work in the cluster but that have group UICs unique to their local computer. As a rule, the group UIC for any given work category should be the same on each computer in the cluster. For example, data entry accounts on VENUS should have the same group UIC as data entry accounts on MARS.<br>**Note:** If you change the UIC for a particular user, you should also change the owner UICs for that user's existing files and directories. You can use the DCL commands SET FILE and SET DIRECTORY to make these changes. These commands are described in detail in the *OpenVMS DCL Dictionary*. |
| **3** | Choose the SYSUAF.DAT file from one of the computers to be a master SYSUAF.DAT.<br>**Note:** The default values for a number of SYSUAF process limits and quotas are higher on an Alpha computer than they are on a VAX computer. See *A Comparison of System Management on OpenVMS AXP and OpenVMS VAX*[1] for information about setting values on both computers. |

[1]This manual has been archived but is available in PostScript and DECW$BOOK (Bookreader) formats on the OpenVMS Documentation CD–ROM. A printed book cn be ordered through DECdirect (800-354-4825).

(continued on next page)

**Table B–1 (Cont.) Building a Common SYSUAF.DAT File**

| Step | Action |
|---|---|
| **4** | Merge the SYSUAF.DAT files from the other computers to the master SYSUAF.DAT by running the Convert utility (CONVERT) on the computer that owns the master SYSUAF.DAT. (See the *OpenVMS Record Management Utilities Reference Manual* for a description of CONVERT.) To use CONVERT to merge the files, each SYSUAF.DAT file must be accessible to the computer that is running CONVERT.<br><br>**Syntax:** To merge the UAFs into the master SYSUAF.DAT file, specify the CONVERT command in the following format:<br><br>`CONVERT SYSUAF1,SYSUAF2,...SYSUAFn  MASTER_SYSUAF`<br><br>Note that if a given user name appears in more than one source file, only the first occurrence of that name appears in the merged file.<br><br>**Example:** The following command sequence example creates a new SYSUAF.DAT file from the combined contents of the two input files:<br><br>`$ SET DEFAULT SYS$SYSTEM`<br>`$ CONVERT/MERGE [SYS1.SYSEXE]SYSUAF.DAT, -`<br>`_$ [SYS2.SYSEXE]SYSUAF.DAT SYSUAF.DAT`<br><br>The CONVERT command in this example adds the records from the files [SYS1.SYSEXE]SYSUAF.DAT and [SYS2.SYSEXE]SYSUAF.DAT to the file SYSUAF.DAT on the local computer.<br><br>After you run CONVERT, you have a master SYSUAF.DAT that contains records from the other SYSUAF.DAT files. |
| **5** | Use AUTHORIZE to modify the accounts in the master SYSUAF.DAT according to the changes you marked on the initial listings of the SYSUAF.DAT files from each computer. |
| **6** | Place the master SYSUAF.DAT file in SYS$COMMON:[SYSEXE]. |
| **7** | Remove all node-specific SYSUAF.DAT files. |

## B.2 Merging RIGHTSLIST.DAT Files

If you need to merge RIGHTSLIST.DAT files, you can use a command sequence like the following:

```
$ ACTIVE_RIGHTSLIST = F$PARSE("RIGHTSLIST","SYS$SYSTEM:.DAT")
$ CONVERT/SHARE/STAT 'ACTIVE_RIGHTSLIST' RIGHTSLIST.NEW
$ CONVERT/MERGE/STAT/EXCEPTION=RIGHTSLIST_DUPLICATES.DAT  -
_$ [SYS1.SYSEXE]RIGHTSLIST.DAT, [SYS2.SYSEXE]RIGHTSLIST.DAT RIGHTSLIST.NEW
$ DUMP/RECORD RIGHTSLIST_DUPLICATES.DAT
$ CONVERT/NOSORT/FAST/STAT RIGHTSLIST.NEW 'ACTIVE_RIGHTSLIST'
```

The commands in this example add the RIGHTSLIST.DAT files from two OpenVMS Cluster computers to the master RIGHTSLIST.DAT file in the current default directory. For detailed information about creating and maintaining RIGHTSLIST.DAT files, see the security guide for your system.

# C
# Cluster Troubleshooting

## C.1 Diagnosing Computer Failures

This appendix contains information to help you perform troubleshooting operations for the following:

- Failures of computers to boot or to join the cluster
- Cluster hangs
- CLUEXIT bugchecks
- Port device problems

### C.1.1 Preliminary Checklist

Before you initiate diagnostic procedures, be sure to verify that these conditions are met:

- All cluster hardware components are correctly connected and checked for proper operation.
- When you attempt to add a new or recently repaired CI computer to the cluster, verify that the CI cables are correctly connected, as described in Section C.10.5.
- OpenVMS Cluster computers and mass storage devices are configured according to requirements specified in the OpenVMS Cluster Software *Software Product Description* (SPD 29.78.*xx*).
- When attempting to add a satellite to a cluster, you must verify that the LAN is configured according to requirements specified in the OpenVMS Cluster Software SPD. You must also verify that you have correctly configured and started the network, following the procedures described in Chapter 4.

If, after performing preliminary checks and taking appropriate corrective action, you find that a computer still fails to boot or to join the cluster, you can follow the procedures in Sections C.2 through C.4 to attempt recovery.

### C.1.2 Sequence of Booting Events

To perform diagnostic and recovery procedures effectively, you must understand the events that occur when a computer boots and attempts to join the cluster. This section outlines those events and shows typical messages displayed at the console.

Note that events vary, depending on whether a computer is the first to boot in a new cluster or whether it is booting in an active cluster. Note also that some events (such as loading the cluster database containing the password and group number) occur only in OpenVMS Cluster systems on a LAN.

The normal sequence of events is shown in Table C–1.

**Table C–1 Sequence of Booting Events**

| Step | Action |
|---|---|
| **1** | The computer boots. If the computer is a satellite, a message like the following shows the name and LAN address of the MOP server that has downline loaded the satellite. At this point, the satellite has completed communication with the MOP server and further communication continues with the system disk server, using OpenVMS Cluster communications.<br><br>`%VAXcluster-I-SYSLOAD, system loaded from Node X...`<br><br>For any booting computer, the OpenVMS "banner message" is displayed in the following format:<br><br>*operating-system* Version *n.n dd-mmm-yyyy hh:mm.ss* |
| **2** | The computer attempts to form or join the cluster, and the following message appears:<br><br>`waiting to form or join an OpenVMS Cluster system`<br><br>If the computer is a member of an OpenVMS Cluster based on the LAN, the cluster security database (containing the cluster password and group number) is loaded. Optionally, the MSCP server and TMSCP server can be loaded:<br><br>`%VAXcluster-I-LOADSECDB, loading the cluster security database`<br>`%MSCPLOAD-I-LOADMSCP, loading the MSCP disk server`<br>`%TMSCPLOAD-I-LOADTMSCP, loading the TMSCP tape server` |
| **3** | If the computer discovers a cluster, the computer attempts to join it. If a cluster is found, the connection manager displays one or more messages in the following format:<br><br>`%CNXMAN, Sending VAXcluster membership request to system X...`<br><br>Otherwise, the connection manager forms the cluster when it has enough votes to establish quorum (that is, when enough voting computers have booted). |
| **4** | As the booting computer joins the cluster, the connection manager displays a message in the following format:<br><br>`%CNXMAN, now a VAXcluster member -- system X...`<br><br>Note that if quorum is lost while the computer is booting, or if a computer is unable to join the cluster within 2 minutes of booting, the connection manager displays messages like the following:<br><br>`%CNXMAN, Discovered system X...`<br>`%CNXMAN, Deleting CSB for system X...`<br>`%CNXMAN, Established "connection" to quorum disk`<br>`%CNXMAN, Have connection to system X...`<br>`%CNXMAN, Have "connection" to quorum disk`<br><br>The last two messages show any connections that have already been formed.<br><br>(continued on next page) |

**Table C–1 (Cont.) Sequence of Booting Events**

| Step | Action |
|---|---|
| **5** | If the cluster includes a quorum disk, you may also see messages like the following:<br>`%CNXMAN, Using remote access method for quorum disk`<br>`%CNXMAN, Using local access method for quorum disk`<br>The first message indicates that the connection manager is unable to access the quorum disk directly, either because the disk is unavailable or because it is accessed through the MSCP server. Another computer in the cluster that can access the disk directly must verify that a reliable connection to the disk exists.<br>The second message indicates that the connection manager can access the quorum disk directly and can supply information about the status of the disk to computers that cannot access the disk directly.<br>**Note:** The connection manager may not see the quorum disk initially because the disk may not yet be configured. In that case, the connection manager first uses remote access, then switches to local access. |
| **6** | Once the computer has joined the cluster, normal startup procedures execute. One of the first functions is to start the OPCOM process:<br>`%%%%%%%%%%  OPCOM  15-JAN-1994 16:33:55.33  %%%%%%%%%%`<br>`Logfile has been initialized by operator _X...$OPA0:`<br>`Logfile is SYS$SYSROOT:[SYSMGR]OPERATOR.LOG;17`<br>`%%%%%%%%%%  OPCOM  15-JAN-1994 16:33:56.43  %%%%%%%%%%`<br>`16:32:32.93  Node X... (csid 0002000E) is now a VAXcluster member` |
| **7** | As other computers join the cluster, OPCOM displays messages like the following:<br>`%%%%%  OPCOM  15-JAN-1994 16:34:25.23  %%%%%  (from node X...)`<br>`16:34:24.42 Node X... (csid 000100F3)`<br>`received VAXcluster membership request from node X...` |

As startup procedures continue, various messages report startup events.

**Hint:** For troubleshooting purposes, you can include in your site-specific startup procedures messages announcing each phase of the startup process—for example, mounting disks or starting queues.

## C.2 Computer on the CI Fails to Boot

If a CI computer fails to boot, perform the following checks:

| Step | Action |
|---|---|
| **1** | Verify that the computer's SCSNODE and SCSSYSTEMID parameters are unique in the cluster. If they are not, you must either alter *both* values or reboot all other computers. |
| **2** | Verify that you are using the correct bootstrap command file. This file must specify the internal bus computer number (if applicable), the HSC or HSJ node number, and the disk from which the computer is to boot. Refer to your processor-specific installation and operations guide for information about setting values in default bootstrap command procedures. |
| **3** | Verify that the PAMAXPORT system parameter is set to a value greater than or equal to the largest CI port number. |
| **4** | Verify that the CI port has a unique hardware station address. |
| **5** | Verify that the HSC subsystem is on line. The ONLINE switch on the HSC operator control panel should be pressed in. |
| **6** | Verify that the disk is available. The correct port switches on the disk's operator control panel should be pressed in. |

| Step | Action |
|---|---|
| **7** | Verify that the computer has access to the HSC subsystem. The SHOW HOSTS command of the HSC SETSHO utility displays status for all computers (hosts) in the cluster. If the computer in question appears in the display as DISABLED, use the SETSHO utility to set the computer to the ENABLED state.<br>**Reference:** For complete information about the SETSHO utility, consult the HSC hardware documentation. |
| **8** | Verify that the HSC subsystem allows access to the boot disk. Invoke the SETSHO utility to ensure that the boot disk is available to the HSC subsystem. The utility's SHOW DISKS command displays the current state of all disks visible to the HSC subsystem and displays all disks in the no-host-access table. |

| IF... | THEN... |
|---|---|
| The boot disk appears in the no-host-access table. | Use the SETSHO utility to set the boot disk to host-access. |
| The boot disk is available or mounted and host access is enabled, but the disk does not appear in the no-host-access table. | Contact your support representative and explain both the problem and the steps you have taken. |

## C.3 Satellite Fails to Boot

To boot successfully, a satellite must communicate with a MOP server over the LAN. You can use DECnet event logging to verify this communication. Proceed as follows:

| Step | Action |
|---|---|
| **1** | Log in as system manager on the MOP server. |
| **2** | If event logging for management-layer events is not already enabled, enter the following NCP commands to enable it:<br>`NCP> SET LOGGING MONITOR EVENT 0.*`<br>`NCP> SET LOGGING MONITOR STATE ON` |
| **3** | Enter the following DCL command to enable the terminal to receive DECnet messages reporting downline load events:<br>`$ REPLY/ENABLE=NETWORK` |

| Step | Action |
|---|---|
| **4** | Boot the satellite. If the satellite and the MOP server can communicate and all boot parameters are correctly set, messages like the following are displayed at the MOP server's terminal: |

```
DECnet event 0.3, automatic line service
From node 2.4 (URANUS), 15-JAN-1994 09:42:15.12
Circuit QNA-0, Load, Requested, Node = 2.42 (OBERON)
File = SYS$SYSDEVICE:<SYS10.>, Operating system
Ethernet address = 08-00-2B-07-AC-03
```

```
DECnet event 0.3, automatic line service
From node 2.4 (URANUS), 15-JAN-1994 09:42:16.76
Circuit QNA-0, Load, Successful, Node = 2.44 (ARIEL)
File = SYS$SYSDEVICE:<SYS11.>, Operating system
Ethernet address = 08-00-2B-07-AC-13
```

| WHEN... | THEN... |
|---|---|
| The satellite cannot communicate with the MOP server (VAX or Alpha). | No message for that satellite appears. There may be a problem with a LAN cable connection or adapter service. |
| The satellite's data in the DECnet database is incorrectly specified (for example, if the hardware address is incorrect). | A message like the following displays the correct address and indicates that a load was requested: |

```
DECnet event 0.7, aborted service
                                    request
From node 2.4 (URANUS) 15-JAN-1994
Circuit QNA-0, Line open error
Ethernet address=08-00-2B-03-29-99
```

Note the absence of the node name, node address, and system root.

Sections C.3.2 through C.3.5 provide more information about satellite boot troubleshooting and often recommend that you ensure that the system parameters are set correctly.

## C.3.1 Displaying Connection Messages

To enable the display of connection messages during a conversational boot, perform the following steps:

| Step | Action |
|---|---|
| **1** | Enable conversational booting by setting the satellite's NISCS_CONV_BOOT system parameter to 1. On Alpha systems, update the ALPHAVMSSYS.PAR file, and on VAX systems update the VAXVMSSYS.PAR file in the system root on the disk server. |
| **2** | Perform a conversational boot. |

‡On Alpha systems, enter the following command at the console:

```
>>>  b -flags 0,1
```

†On VAX systems, set bit <0> in register R5. For example, on a VAXstation 3100 system, enter the following command on the console:

```
>>>  B/1
```

†VAX specific

‡Alpha specific

| Step | Action |
|---|---|
| **3** | Observe connection messages.<br><br>Display connection messages during a satellite boot to determine which system in a large cluster is serving the system disk to a cluster satellite during the boot process. If booting problems occur, you can use this display to help isolate the problem with the system that is currently serving the system disk. Then, if your server system has multiple LAN adapters, you can isolate specific LAN adapters. |
| **4** | Isolate LAN adapters.<br><br>Isolate a LAN adapter by methodically rebooting with only one adapter connected. That is, disconnect all but one of the LAN adapters on the server system and reboot the satellite. If the satellite boots when it is connected to the system disk server, then follow the same procedure using a different LAN adapter. Continue these steps until you have located the bad adapter. |

**Reference:** See also Appendix C for help with troubleshooting satellite booting problems.

### C.3.2 General OpenVMS Cluster Satellite-Boot Troubleshooting

If a satellite fails to boot, use the steps outlined in this section to diagnose and correct problems in OpenVMS Cluster systems.

| Step | Action |
|---|---|
| **1** | Verify that the boot device is available. This check is particularly important for clusters in which satellites boot from multiple system disks. |
| **2** | Verify that the DECnet network is up and running. |
| **3** | Check the cluster group code and password. The cluster group code and password are set using the CLUSTER_CONFIG.COM procedure. |
| **4** | Verify that you have installed the correct OpenVMS Alpha and OpenVMS VAX licenses. |
| **5** | Verify system parameter values on each satellite node, as follows:<br><br>VAXCLUSTER = 2<br>NISCS_LOAD_PEA0 = 1<br>NISCS_LAN_OVRHD = 0<br>NISCS_MAX_PKTSZ = 1498[1]<br>SCSNODE is the name of the computer.<br>SCSSYSTEMID is a number that identifies the computer.<br>VOTES = 0<br><br>The SCS parameter values are set differently depending on your system configuration.<br><br>**Reference:** Appendix A describes how to set these SCS parameters. |

[1]For Ethernet adapters, the value of NISCS_MAX_PKTSZ is 1498. For FDDI adapters, the value is 4468.

| Step | Action |
|---|---|

To check system parameter values on a satellite node that cannot boot, invoke the SYSGEN utility on a running system in the OpenVMS Cluster that has access to the satellite node's local root. (Note that you must invoke the SYSGEN utility from a node that is running the same type of operating system—for example, to troubleshoot an Alpha satellite node, you must run the SYSGEN utility on an Alpha system.) Check system parameters as follows:

| Step | Action |
|---|---|
| A | Find the local root of the satellite node on the system disk. The following example is from an Alpha system running DECnet for OpenVMS: |

```
$ MCR NCP SHOW NODE HOME CHARACTERISTICS

Node Volatile Characteristics as of 10-JAN-1994 09:32:56

Remote node =   63.333 (HOME)

Hardware address         = 08-00-2B-30-96-86
Load file                = APB.EXE
Load Assist Agent        = SYS$SHARE:NISCS_LAA.EXE
Load Assist Parameter    = ALPHA$SYSD:[SYS17.]
```

The local root in this example is ALPHA$SYSD:[SYS17.].

**Reference:** Refer to the DECnet–Plus documentation for equivalent information using NCL commands.

B Enter the SHOW LOGICAL command at the system prompt to translate the logical name for ALPHA$SYSD.

```
$ SHO LOG ALPHA$SYSD
   "ALPHA$SYSD" = "$69$DUA121:" (LNM$SYSTEM_TABLE)
```

C Invoke the SYSGEN utility on the system from which you can access the satellite's local disk. (This example invokes the SYSGEN utility on an Alpha system using the Alpha parameter file ALPHAVMSSYS.PAR. The SYSGEN utility on VAX systems differs in that it uses the VAX parameter file VAXVMSSYS.PAR). The following example illustrates how to enter the SYSGEN command USE with the system parameter file on the local root for the satellite node and then enter the SHOW command to query the parameters in question.

```
$  MCR SYSGEN

SYSGEN> USE $69$DUA121:[SYS17.SYSEXE]ALPHAVMSSYS.PAR
SYSGEN> SHOW VOTES
Parameter
Name      Current Default Min. Max. Unit Dynamic
--------- ------- ------- --- ----- ---- -------
VOTES           0       1   0   127 Votes
SYSGEN> EXIT
```

### C.3.3 MOP Server Troubleshooting

To diagnose and correct problems for MOP servers, follow the steps outlined in this section.

| Step | Action |
|---|---|
| **1** | Perform the steps outlined in Section C.3.2. |
| **2** | Verify the NCP circuit state is on and the service is enabled. Enter the following commands to run the NCP utility and check the NCP circuit state.<br><br>`$ MCR NCP`<br>`NCP> SHOW CIRCUIT ISA-0 CHARACTERISTICS`<br><br>`Circuit Volatile Characteristics as of 12-JAN-1994 10:08:30`<br><br>`Circuit = ISA-0`<br><br>`State = on`<br>`Service = enabled`<br>`Designated router = 63.1021`<br>`Cost = 10`<br>`Maximum routers allowed = 33`<br>`Router priority = 64`<br>`Hello timer = 15`<br>`Type = Ethernet`<br>`Adjacent node = 63.1021`<br>`Listen timer = 45` |
| **3** | If service is not enabled, you can enter NCP commands like the following to enable it:<br><br>`NCP> SET CIRCUIT` *`circuit-id`* `STATE OFF`<br>`NCP> DEFINE CIRCUIT` *`circuit-id`* `SERVICE ENABLED`<br>`NCP> SET CIRCUIT` *`circuit-id`* `SERVICE ENABLED STATE ON`<br><br>The DEFINE command updates the permanent database and ensures that service is enabled the next time you start the network. Note that DECnet traffic is interrupted while the circuit is off. |
| **4** | Verify that the load assist parameter points to the system disk and the system root for the satellite. |
| **5** | Verify that the satellite's system disk is mounted on the MOP server node. |
| **6** | ‡On Alpha systems, verify that the load file is APB.EXE. |
| **7** | For MOP booting, the satellite node's parameter file (ALPHAVMSYS.PAR for Alpha computers and VAXVMSSYS.PAR for VAX computers) must be located in the [SYSEXE] directory of the satellite system root. |
| **8** | Ensure that the file CLUSTER_AUTHORIZE.DAT is located in the [SYSCOMMON.SYSEXE] directory of the satellite system root. |

‡Alpha specific

### C.3.4 Disk Server Troubleshooting

To diagnose and correct problems for disk servers, follow the steps outlined in this section.

| Step | Action |
|---|---|
| **1** | Perform the steps in Section C.3.2. |
| **2** | For each satellite node, verify the following system parameter values:<br>MSCP_LOAD = 1<br>MSCP_SERVE_ALL = 1 |
| **3** | The disk servers for the system disk must be connected directly to the disk. |

### C.3.5 Satellite Booting Troubleshooting

To diagnose and correct problems for satellite booting, follow the steps outlined in this section.

| Step | Action |
|---|---|
| **1** | Perform the steps in Sections C.3.2, C.3.3, and C.3.4. |
| **2** | For each satellite node, verify that the VOTES system parameter is set to 0. |
| **3** | ‡On Alpha systems, verify the DECnet network database on the MOP servers by running the NCP utility and entering the following commands to display node characteristics. The following example displays information about an Alpha node named UTAH:<br>`$ MCR NCP`<br>`NCP> SHOW NODE UTAH CHARACTERISTICS`<br>`Node Volatile Characteristics as of 15-JAN-1994 10:28:09`<br>`Remote node = 63.227 (UTAH)`<br>`Hardware address = 08-00-2B-2C-CE-E3`<br>`Load file = APB.EXE`<br>`Load Assist Agent = SYS$SHARE:NISCS_LAA.EXE`<br>`Load Assist Parameter = $69$DUA100:[SYS17.]`<br>The load file must be APB.EXE. In addition, when booting Alpha nodes, for each LAN adapter specified on the boot command line, the load assist parameter must point to the same system disk and root number. |
| **4** | †On VAX systems, verify the DECnet network database on the MOP servers by running the NCP utility and entering the following commands to display node characteristics. The following example displays information about a VAX node named ARIEL:<br>`$ MCR NCP`<br>`NCP> SHOW CHAR NODE ARIEL`<br>`Node Volatile Characteristics as of 15-JAN-1994 13:15:28`<br>`Remote node = 2.41 (ARIEL)`<br>`Hardware address = 08-00-2B-03-27-95`<br>`Tertiary loader = SYS$SYSTEM:TERTIARY_VMB.EXE`<br>`Load Assist Agent = SYS$SHARE:NISCS_LAA.EXE`<br>`Load Assist Parameter = DISK$VAXVMSRL5:<SYS12.>`<br>Note that on VAX nodes, the tertiary loader is SYS$SYSTEM:TERTIARY_VMB.EXE. |

†VAX specific

‡Alpha specific

| Step | Action |
|---|---|
| **5** | On Alpha and VAX systems, verify the following information in the NCP display: |

| Step | Action |
|---|---|
| A | Verify the DECnet address for the node. |
| B | Verify the load assist agent is SYS$SHARE:NISCS_LAA.EXE. |
| C | Verify the load assist parameter points to the satellite system disk and correct root. |
| D | Verify that the hardware address matches the satellite's Ethernet address. At the satellite's console prompt, use the information shown in Table 8–3 to obtain the satellite's current LAN hardware address.<br><br>Compare the hardware address values displayed by NCP and at the satellite's console. The values should be identical and should also match the value shown in the SYS$MANAGER:NETNODE_UPDATE.COM file. If the values do not match, you must make appropriate adjustments. For example, if you have recently replaced the satellite's LAN adapter, you must execute CLUSTER_CONFIG.COM CHANGE function to update the network database and NETNODE_UPDATE.COM on the appropriate MOP server. |

| | |
|---|---|
| **6** | Perform a conversational boot to determine more precisely why the satellite is having trouble booting. The conversational boot procedure displays messages that can help you solve network booting problems. The messages provide information about the state of the network and the communications process between the satellite and the system disk server.<br><br>**Reference:** Section C.3.6 describes booting messages for Alpha systems. |

## C.3.6 Alpha Booting Messages (Alpha Only)

On Alpha systems, the messages are displayed as shown in Table C–2.

**Table C–2 Alpha Booting Messages (Alpha Only)**

| Message | Comments |
|---|---|
| **%VMScluster-I-MOPSERVER, MOP server for downline load was node UTAH** | |
| This message displays the name of the system providing the DECnet MOP downline load. This message acknowledges that control was properly transferred from the console performing the MOP load to the image that was loaded. | If this message is not displayed, either the MOP load failed or the wrong file was MOP downline loaded. |
| **%VMScluster-I-BUSONLINE, LAN adapter is now running 08-00-2B-2C-CE-E3** | |
| This message displays the LAN address of the Ethernet or FDDI adapter specified in the boot command. Multiple lines can be displayed if multiple LAN devices were specified in the boot command line. The booting satellite can now attempt to locate the system disk by sending a message to the cluster multicast address. | If this message is not displayed, the LAN adapter is not initialized properly. Check the physical network connection. For FDDI, the adapter must be on the ring. |

(continued on next page)

**Table C–2 (Cont.) Alpha Booting Messages (Alpha Only)**

| Message | Comments |
|---|---|
| **%VMScluster-I-VOLUNTEER, System disk service volunteered by node EUROPA AA-00-04-00-4C-FD** | |
| This message displays the name of a system claiming to serve the satellite system disk. This system has responded to the multicast message sent by the booting satellite to locate the servers of the system disk. | If this message is not displayed, one or more of the following situations may be causing the problem:<br>– The network path between the satellite and the boot server either is broken or is filtering the local area OpenVMS Cluster multicast messages.<br>– The system disk is not being served.<br>– The CLUSTER_AUTHORIZE.DAT file on the system disk does not match the other cluster members. |
| **%VMScluster-I-CREATECH, Creating channel to node EUROPA 08-00-2B-2C-CE-E2 08-00-2B-12-AE-A2** | |
| This message displays the LAN address of the local LAN adapter (first address) and of the remote LAN adapter (second address) that form a communications path through the network. These adapters can be used to support a NISCA virtual circuit for booting. Multiple messages can be displayed if either multiple LAN adapters were specified on the boot command line or the system serving the system disk has multiple LAN adapters. | If you do not see as many of these messages as you expect, there may be network problems related to the LAN adapters whose addresses are not displayed. Use the Local Area OpenVMS Cluster Network Failure Analysis Program for better troubleshooting (see Section D.5). |
| **%VMScluster-I-OPENVC, Opening virtual circuit to node EUROPA** | |
| This message displays the name of a system that has established an NISCA virtual circuit to be used for communications during the boot process. Booting uses this virtual circuit to connect to the remote MSCP server. | |
| **%VMScluster-I-MSCPCONN, Connected to a MSCP server for the system disk, node EUROPA** | |
| This message displays the name of a system that is actually serving the satellite system disk. | If this message is not displayed, the system that claimed to serve the system disk could not serve the disk. Check the OpenVMS Cluster configuration. |
| **%VMScluster-W-SHUTDOWNCH, Shutting down channel to node EUROPA 08-00-2B-2C-CE-E3 08-00-2B-12-AE-A2** | |
| This message displays the LAN address of the local LAN adapter (first address) and of the remote LAN adapter (second address) that have just lost communications. Depending on the type of failure, multiple messages may be displayed if either the booting system or the system serving the system disk has multiple LAN adapters. | |
| **%VMScluster-W-CLOSEVC, Closing virtual circuit to node EUROPA** | |
| This message indicates that NISCA communications have failed to the system whose name is displayed. | |

(continued on next page)

**Table C–2 (Cont.) Alpha Booting Messages (Alpha Only)**

| Message | Comments |
|---|---|
| **%VMScluster-I-RETRY, Attempting to reconnect to a system disk server** | |
| This message indicates that an attempt will be made to locate another system serving the system disk. The LAN adapters will be reinitialized and all communications will be restarted. | |
| **%VMScluster-W-PROTOCOL_TIMEOUT, NISCA protocol timeout** | |
| Either the booting node has lost connections to the remote system or the remote system is no longer responding to requests made by the booting system. In either case, the booting system has declared a failure and will reestablish communications to a boot server. | |

## C.4 Computer Fails to Join the Cluster

If a computer fails to join the cluster, follow the procedures in this section to determine the cause.

### C.4.1 Verifying OpenVMS Cluster Software Load

To verify that OpenVMS Cluster software has been loaded, follow these instructions:

| Step | Action |
|---|---|
| **1** | Look for connection manager (%CNXMAN) messages like those shown in Section C.1.2. |
| **2** | If no such messages are displayed, OpenVMS Cluster software probably was not loaded at boot time. Reboot the computer in conversational mode. At the SYSBOOT> prompt, set the VAXCLUSTER parameter to 2. |
| **3** | For OpenVMS Cluster systems communicating over the LAN or mixed interconnects, set NISCS_LOAD_PEA0 to 1 and VAXCLUSTER to 2. These parameters should also be set in the computer's MODPARAMS.DAT file. (For more information about booting a computer in conversational mode, consult your installation and operations guide). |
| **4** | For OpenVMS Cluster systems on the LAN, verify that the cluster security database file (SYS$COMMON:CLUSTER_AUTHORIZE.DAT) exists and that you have specified the correct group number for this cluster (see Section 10.9.1). |

### C.4.2 Verifying Boot Disk and Root

To verify that the computer has booted from the correct disk and system root, follow these instructions:

| Step | Action |
|---|---|
| **1** | If %CNXMAN messages are displayed, and if, after the conversational reboot, the computer still does not join the cluster, check the console output on all active computers and look for messages indicating that one or more computers found a remote computer that conflicted with a known or local computer. Such messages suggest that two computers have booted from the same system root. |
| **2** | Review the boot command files for all CI computers and ensure that all are booting from the correct disks and from unique system roots. |

| Step | Action |
|---|---|
| 3 | If you find it necessary to modify the computer's bootstrap command procedure (console media), you may be able to do so on another processor that is already running in the cluster.<br><br>Replace the running processor's console media with the media to be modified, and use the Exchange utility and a text editor to make the required changes. Consult the appropriate processor-specific installation and operations guide for information about examining and editing boot command files. |

### C.4.3 Verifying SCSNODE and SCSSYSTEMID Parameters

To be eligible to join a cluster, a computer must have unique SCSNODE and SCSSYSTEMID parameter values.

| Step | Action |
|---|---|
| 1 | Check that the current values do not duplicate any values set for existing OpenVMS Cluster computers. To check values, you can perform a conversational bootstrap operation. |
| 2 | If the values of SCSNODE or SCSSYSTEMID are not unique, do either of the following:<br><br>• Alter *both* values.<br><br>• Reboot all other computers.<br><br>**Note:** To modify values, you can perform a conversational bootstrap operation. However, for reliable future bootstrap operations, specify appropriate values for these parameters in the computer's MODPARAMS.DAT file. |

| WHEN you change... | THEN... |
|---|---|
| The SCSNODE parameter | Change the DECnet node name too, because both names must be the same. |
| Either the SCSNODE parameter or the SCSSYSTEMID parameter on a node that was previously an OpenVMS Cluster member | Change the DECnet node number, too, because both numbers must be the same. Reboot the entire cluster. |

### C.4.4 Verifying Cluster Security Information

To verify the cluster group code and password, follow these instructions:

| Step | Action |
|---|---|
| 1 | Verify that the database file SYS$COMMON:CLUSTER_AUTHORIZE.DAT exists. |
| 2 | For clusters with multiple system disks, ensure that the correct (same) group number and password were specified for each.<br><br>**Reference:** See Section 10.9 to view the group number and to reset the password in the CLUSTER_AUTHORIZE.DAT file using the SYSMAN utility. |

## C.5 Startup Procedures Fail to Complete

If a computer boots and joins the cluster but appears to hang before startup procedures complete—that is, before you are able to log in to the system—be sure that you have allowed sufficient time for the startup procedures to execute.

| IF... | THEN... |
|---|---|
| The startup procedures fail to complete after a period that is normal for your site. | Try to access the procedures from another OpenVMS Cluster computer and make appropriate adjustments. For example, verify that all required devices are configured and available. One cause of such a failure could be the lack of some system resource, such as NPAGEDYN or page file space. |
| You suspect that the value for the NPAGEDYN parameter is set too low. | Perform a conversational bootstrap operation to increase it. Use SYSBOOT to check the current value, and then double the value. |
| You suspect a shortage of page file space, and another OpenVMS Cluster computer is available. | Log in on that computer and use the System Generation utility (SYSGEN) to provide adequate page file space for the problem computer.<br>**Note:** Insufficent page-file space on the booting computer might cause other computers to hang. |
| The computer still cannot complete the startup procedures. | Contact your Compaq support representative. |

## C.6 Diagnosing LAN Component Failures

Section D.5 provides troubleshooting techniques for LAN component failures (for example, broken LAN bridges). That appendix also describes techniques for using the Local Area OpenVMS Cluster Network Failure Analysis Program.

Intermittent LAN component failures (for example, packet loss) can cause problems in the NISCA transport protocol that delivers System Communications Services (SCS) messages to other nodes in the OpenVMS Cluster. Appendix F describes troubleshooting techniques and requirements for LAN analyzer tools.

## C.7 Diagnosing Cluster Hangs

Conditions like the following can cause a OpenVMS Cluster computer to suspend process or system activity (that is, to hang):

| Condition | Reference |
|---|---|
| Cluster quorum is lost. | Section C.7.1 |
| A shared cluster resource is inaccessible. | Section C.7.2 |

### C.7.1 Cluster Quorum is Lost

The OpenVMS Cluster quorum algorithm coordinates activity among OpenVMS Cluster computers and ensures the integrity of shared cluster resources. (The quorum algorithm is described fully in Chapter 2.) Quorum is checked after any change to the cluster configuration—for example, when a voting computer leaves or joins the cluster. If quorum is lost, process and I/O activity on all computers in the cluster are blocked.

Information about the loss of quorum and about clusterwide events that cause loss of quorum are sent to the OPCOM process, which broadcasts messages to designated operator terminals. The information is also broadcast to each computer's operator console (OPA0), unless broadcast activity is explicitly disabled on that terminal. However, because quorum may be lost before OPCOM has been able to inform the operator terminals, the messages sent to OPA0 are the most reliable source of information about events that cause loss of quorum.

If quorum is lost, you might add or reboot a node with additional votes.

**Reference:** See also the information about cluster quorum in Section 10.12.

### C.7.2 Inaccessible Cluster Resource

Access to shared cluster resources is coordinated by the distributed lock manager. If a particular process is granted a lock on a resource (for example, a shared data file), other processes in the cluster that request incompatible locks on that resource must wait until the original lock is released. If the original process retains its lock for an extended period, other processes waiting for the lock to be released may appear to hang.

Occasionally, a system activity must acquire a restrictive lock on a resource for an extended period. For example, to perform a volume rebuild, system software takes out an exclusive lock on the volume being rebuilt. While this lock is held, no processes can allocate space on the disk volume. If they attempt to do so, they may appear to hang.

Access to files that contain data necessary for the operation of the system itself is coordinated by the distributed lock manager. For this reason, a process that acquires a lock on one of these resources and is then unable to proceed may cause the cluster to appear to hang.

For example, this condition may occur if a process locks a portion of the system authorization file (SYS$SYSTEM:SYSUAF.DAT) for write access. Any activity that requires access to that portion of the file, such as logging in to an account with the same or similar user name or sending mail to that user name, is blocked until the original lock is released. Normally, this lock is released quickly, and users do not notice the locking operation.

However, if the process holding the lock is unable to proceed, other processes could enter a wait state. Because the authorization file is used during login and for most process creation operations (for example, batch and network jobs), blocked processes could rapidly accumulate in the cluster. Because the distributed lock manager is functioning normally under these conditions, users are not notified by broadcast messages or other means that a problem has occurred.

## C.8 Diagnosing CLUEXIT Bugchecks

The operating system performs **bugcheck** operations only when it detects conditions that could compromise normal system activity or endanger data integrity. A **CLUEXIT bugcheck** is a type of bugcheck initiated by the connection manager, the OpenVMS Cluster software component that manages the interaction of cooperating OpenVMS Cluster computers. Most such bugchecks are triggered by conditions resulting from hardware failures (particularly failures in communications paths), configuration errors, or system management errors.

### C.8.1 Conditions Causing Bugchecks

The most common conditions that result in CLUEXIT bugchecks are as follows:

| Possible Bugcheck Causes | Recommendations |
|---|---|
| The cluster connection between two computers is broken for longer than RECNXINTERVAL seconds. Thereafter, the connection is declared irrevocably broken. If the connection is later reestablished, one of the computers shut down with a CLUEXIT bugcheck.<br>This condition can occur:<br>• Upon recovery with battery backup after a power failure<br>• After the repair of an SCS communication link<br>• After the computer was halted for a period longer than the number of seconds specified for the RECNXINTERVAL parameter and was restarted with a CONTINUE command entered at the operator console | Determine the cause of the interrupted connection and correct the problem. For example, if recovery from a power failure is longer than RECNXINTERVAL seconds, you may want to increase the value of the RECNXINTERVAL parameter on all computers. |
| Cluster partitioning occurs. A member of a cluster discovers or establishes connection to a member of another cluster, or a foreign cluster is detected in the quorum file. | Review the setting of EXPECTED_VOTES on all computers. |
| The value specified for the SCSMAXMSG system parameter on a computer is too small. | Verify that the value of SCSMAXMSG on all OpenVMS Cluster computers is set to a value that is at the least the default value. |

## C.9 Port Communications

These sections provide detailed information about port communications to assist in diagnosing port communication problems.

### C.9.1 Port Polling

Shortly after a CI computer boots, the CI port driver (PADRIVER) begins configuration polling to discover other active ports on the CI. Normally, the poller runs every 5 seconds (the default value of the PAPOLLINTERVAL system parameter). In the first polling pass, all addresses are probed over cable path A; on the second pass, all addresses are probed over path B; on the third pass, path A is probed again; and so on.

The poller probes by sending Request ID (REQID) packets to all possible port numbers, including itself. Active ports receiving the REQIDs return ID Received packet (IDREC) to the port issuing the REQID. A port might respond to a REQID even if the computer attached to the port is not running.

For OpenVMS Cluster systems communicating over the CI, DSSI, or a combination of these interconnects, the port drivers perform a start handshake when a pair of ports and port drivers has successfully exchanged ID packets. The port drivers exchange datagrams containing information about the computers, such as the type of computer and the operating system version. If this exchange is successful, each computer declares a virtual circuit open. An open virtual circuit is prerequisite to all other activity.

### C.9.2 LAN Communications

For clusters that include Ethernet or FDDI interconnects, a multicast scheme is used to locate computers on the LAN. Approximately every 3 seconds, the port emulator driver (PEDRIVER) sends a HELLO datagram message through each LAN adapter to a cluster-specific multicast address that is derived from the cluster group number. The driver also enables the reception of these messages from other computers. When the driver receives a HELLO datagram message from a computer with which it does not currently share an open virtual circuit, it attempts to create a circuit. HELLO datagram messages received from a computer with a currently open virtual circuit indicate that the remote computer is operational.

A standard, three-message exchange handshake is used to create a virtual circuit. The handshake messages contain information about the transmitting computer and its record of the cluster password. These parameters are verified at the receiving computer, which continues the handshake only if its verification is successful. Thus, each computer authenticates the other. After the final message, the virtual circuit is opened for use by both computers.

### C.9.3 System Communications Services (SCS) Connections

System services such as the disk class driver, connection manager, and the MSCP and TMSCP servers communicate between computers with a protocol called System Communications Services (SCS). SCS is responsible primarily for forming and breaking intersystem process connections and for controlling flow of message traffic over those connections. SCS is implemented in the port driver (for example, PADRIVER, PBDRIVER, PEDRIVER, PIDRIVER), and in a loadable piece of the operating system called SCSLOA.EXE (loaded automatically during system initialization).

When a virtual circuit has been opened, a computer periodically probes a remote computer for system services that the remote computer may be offering. The SCS directory service, which makes known services that a computer is offering, is always present both on computers and HSC subsystems. As system services discover their counterparts on other computers and HSC subsystems, they establish SCS connections to each other. These connections are full duplex and are associated with a particular virtual circuit. Multiple connections are typically associated with a virtual circuit.

## C.10 Diagnosing Port Failures

This section describes the hierarchy of communication paths and describes where failures can occur.

### C.10.1 Hierarchy of Communication Paths

Taken together, SCS, the port drivers, and the port itself support a hierarchy of communication paths. Starting with the most fundamental level, these are as follows:

- The physical wires. The Ethernet is a single coaxial cable. FDDI typically has a pair of fiber-optic cables for redundancy. The CI has two pairs of transmitting and receiving cables (path A transmit and receive and path B transmit and receive). For the CI, the operating system software normally sends traffic in automatic path-select mode. The port chooses the free path or, if both are free, an arbitrary path (implemented in the cables and star coupler and managed by the port).

- The virtual circuit (implemented partly in the CI port or LAN port emulator driver (PEDRIVER) and partly in SCS software).
- The SCS connections (implemented in system software).

## C.10.2 Where Failures Occur

Failures can occur at each communication level and in each component. Failures at one level translate into failures elsewhere, as described in Table C–3.

**Table C–3 Port Failures**

| Communication Level | Failures |
|---|---|
| Wires | If the LAN fails or is disconnected, LAN traffic stops or is interrupted, depending on the nature of the failure. For the CI, either path A or B can fail while the virtual circuit remains intact. All traffic is directed over the remaining good path. When the wire is repaired, the repair is detected automatically by port polling, and normal operations resume on all ports. |
| Virtual circuit | If no path works between a pair of ports, the virtual circuit fails and is closed. A path failure is discovered as follows:<br>– For the CI, when polling fails or when attempts are made to send normal traffic, and the port reports that neither path yielded transmit success.<br>– For the LAN, when no multicast HELLO datagram message or incoming traffic is received from another computer.<br>When a virtual circuit fails, every SCS connection on it is closed. The software automatically reestablishes connections when the virtual circuit is reestablished. Normally, reestablishing a virtual circuit takes several seconds after the problem is corrected. |
| CI port | If a port fails, all virtual circuits to that port fail, and all SCS connections on those virtual circuits are closed. If the port is successfully reinitialized, virtual circuits and connections are reestablished automatically. Normally, port reinitialization and reestablishment of connections take several seconds. |
| LAN adapter | If a LAN adapter device fails, attempts are made to restart it. If repeated attempts fail, all channels using that adapter are broken. A channel is a pair of LAN addresses, one local and one remote. If the last open channel for a virtual circuit fails, the virtual circuit is closed and the connections are broken. |
| SCS connection | When the software protocols fail or, in some instances, when the software detects a hardware malfunction, a connection is terminated. Other connections are usually unaffected, as is the virtual circuit. Breaking of connections is also used under certain conditions as an error recovery mechanism—most commonly when there is insufficient nonpaged pool available on the computer. |
| Computer | If a computer fails because of operator shutdown, bugcheck, or halt, all other computers in the cluster record the shutdown as failures of their virtual circuits to the port on the shut down computer. |

## C.10.3 Verifying CI Port Functions

Before you boot in a cluster a CI connected computer that is new, just repaired, or suspected of having a problem, you should have Compaq services verify that the computer runs correctly on its own.

## C.10.4 Verifying Virtual Circuits

To diagnose communication problems, you can invoke the Show Cluster utility using the instructions in Table C–4.

**Table C–4 How to Verify Virtual Circuit States**

| Step | Action | What to Look for |
|---|---|---|
| 1 | Tailor the SHOW CLUSTER report by entering the SHOW CLUSTER command ADD CIRCUIT,CABLE_STATUS. This command adds a class of information about all the virtual circuits as seen from the computer on which you are running SHOW CLUSTER. CABLE_STATUS indicates the status of the path for the circuit from the CI interface on the local system to the CI interface on the remote system. | Primarily, you are checking whether there is a virtual circuit in the OPEN state to the failing computer. Common causes of failure to open a virtual circuit and keep it open are the following:<br>• Port errors on one side or the other<br>• Cabling errors<br>• A port set off line because of software problems<br>• Insufficient nonpaged pool on both sides<br>• Failure to set correct values for the SCSNODE, SCSSYSTEMID, PAMAXPORT, PANOPOLL, PASTIMOUT, and PAPOLLINTERVAL system parameters |
| 2 | Run SHOW CLUSTER from each active computer in the cluster to verify whether each computer's view of the failing computer is consistent with every other computer's view.<br>**WHEN... / THEN...**<br>All the active computers have a consistent view of the failing computer — The problem may be in the failing computer.<br>Only one of several active computers detects that the newcomer is failing — That particular computer may have a problem. | If no virtual circuit is open to the failing computer, check the bottom of the SHOW CLUSTER display:<br>• For information about circuits to the port of the failing computer. Virtual circuits in partially open states are shown at the bottom of the display. If the circuit is shown in a state other than OPEN, communications between the local and remote ports are taking place, and the failure is probably at a higher level than in port or cable hardware.<br>• To see whether both path A and path B to the failing port are good. The loss of one path should not prevent a computer from participating in a cluster. |

## C.10.5 Verifying CI Cable Connections

Whenever the configuration poller finds that no virtual circuits are open and that no handshake procedures are currently opening virtual circuits, the poller analyzes its environment. It does so by using the send-loopback-datagram facility of the CI port in the following fashion:

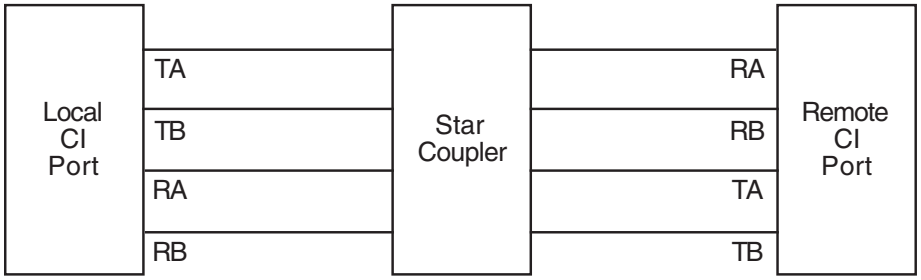
1. The send-loopback-datagram facility tests the connections between the CI port and the star coupler by routing messages across them. The messages are called loopback datagrams. (The port processes other self-directed messages without using the star coupler or external cables.)

2. The configuration poller makes entries in the error log whenever it detects a change in the state of a circuit. Note, however, that it is possible two changed-to-failed-state messages can be entered in the log without an intervening changed-to-succeeded-state message. Such a series of entries means that the circuit state continues to be faulty.

### C.10.6 Diagnosing CI Cabling Problems

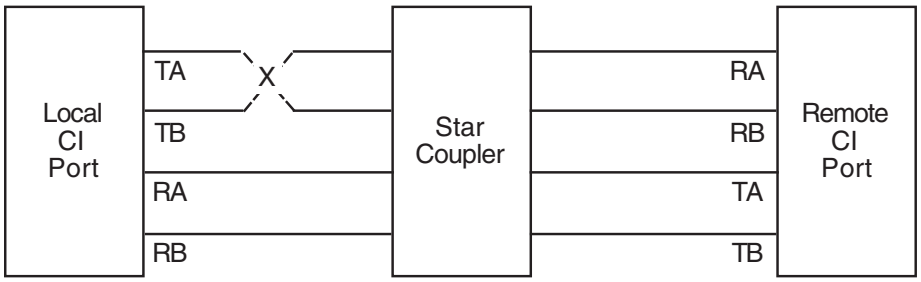
The following paragraphs discuss various incorrect CI cabling configurations and the entries made in the error log when these configurations exist. Figure C–1 shows a two-computer configuration with all cables correctly connected. Figure C–2 shows a CI cluster with a pair of crossed cables.

**Figure C–1 Correctly Connected Two-Computer CI Cluster**




ZK–1924–GE

**Figure C–2 Crossed CI Cable Pair**




ZK–1925–GE

If a pair of transmitting cables or a pair of receiving cables is crossed, a message sent on TA is received on RB, and a message sent on TB is received on RA. This is a hardware error condition from which the port cannot recover. An entry is made in the error log indicating that a single pair of crossed cables exists. The entry contains the following lines:

```
DATA CABLE(S) CHANGE OF STATE
PATH  1.  LOOPBACK HAS GONE FROM GOOD TO BAD
```

If this situation exists, you can correct it by reconnecting the cables properly. The cables could be misconnected in several places. The coaxial cables that connect the port boards to the bulkhead cable connectors can be crossed, or the cables can be misconnected to the bulkhead or the star coupler.

**Configuration 1:** The information illustrated in Figure C–2 is represented more simply in Example C–1. It shows the cables positioned as in Figure C–2, but it does not show the star coupler or the computers. The labels LOC (local) and REM (remote) indicate the pairs of transmitting (T) and receiving (R) cables on the local and remote computers, respectively.

**Example C–1 Crossed Cables: Configuration 1**

```
T x   = R
R =   = T
LOC   REM
```

The pair of crossed cables causes loopback datagrams to fail on the local computer but to succeed on the remote computer. Crossed pairs of transmitting cables and crossed pairs of receiving cables cause the same behavior.

Note that only an odd number of crossed cable pairs causes these problems. If an even number of cable pairs is crossed, communications succeed. An error log entry is made in some cases, however, and the contents of the entry depends on which pairs of cables are crossed.

**Configuration 2:** Example C–2 shows two-computer clusters with the combinations of two crossed cable pairs. These crossed pairs cause the following entry to be made in the error log of the computer that has the cables crossed:

```
DATA CABLE(S) CHANGE OF STATE
CABLES HAVE GONE FROM UNCROSSED TO CROSSED
```

Loopback datagrams succeed on both computers, and communications are possible.

**Example C–2 Crossed Cables: Configuration 2**

```
T x   = R       T =   x R
R x   = T       R =   x T
LOC   REM       LOC   REM
```

**Configuration 3:** Example C–3 shows the possible combinations of two pairs of crossed cables that cause loopback datagrams to fail on both computers in the cluster. Communications can still take place between the computers. An entry stating that cables are crossed is made in the error log of each computer.

**Example C–3 Crossed Cables: Configuration 3**

```
T x   = R       T =   x R
R =   x T       R x   = T
LOC   REM       LOC   REM
```

**Configuration 4:** Example C–4 shows the possible combinations of two pairs of crossed cables that cause loopback datagrams to fail on both computers in the cluster but that allow communications. No entry stating that cables are crossed is made in the error log of either computer.

**Example C–4 Crossed Cables: Configuration 4**

```
T x   x R       T =   = R
R =   = T       R x   x T
LOC   REM       LOC   REM
```

**Configuration 5:** Example C–5 shows the possible combinations of four pairs of crossed cables. In each case, loopback datagrams fail on the computer that has only one crossed pair of cables. Loopback datagrams succeed on the computer with both pairs crossed. No communications are possible.

**Example C–5 Crossed Cables: Configuration 5**

```
T x   x R       T x   = R       T =   x R       T x   x R
R x   = T       R x   x T       R x   x T       R =   x T
LOC   REM       LOC   REM       LOC   REM       LOC   REM
```

If all four cable pairs between two computers are crossed, communications succeed, loopback datagrams succeed, and no crossed-cable message entries are made in the error log. You might detect such a condition by noting error log entries made by a third computer in the cluster, but this occurs only if the third computer has one of the crossed-cable cases described.

## C.10.7 Repairing CI Cables

This section describes some ways in which Compaq support representatives can make repairs on a running computer. This information is provided to aid system managers in scheduling repairs.

For cluster software to survive cable-checking activities or cable-replacement activities, you must be sure that either path A or path B is intact at all times between each port and between every other port in the cluster.

For example, you can remove path A and path B in turn from a particular port to the star coupler. To make sure that the configuration poller finds a path that was previously faulty but is now operational, follow these steps:

| Step | Action |
|---|---|
| **1** | Remove path B. |
| **2** | After the poller has discovered that path B is faulty, reconnect path B. |

| Step | Action |
|---|---|
| **3** | Wait two poller intervals,[1] and then take either of the following actions:<br>• Enter the DCL command SHOW CLUSTER to make sure that the poller has reestablished path B.<br>• Enter the DCL command SHOW CLUSTER/CONTINUOUS followed by the SHOW CLUSTER command ADD CIRCUITS, CABLE_ST. |
| **4** | Wait for SHOW CLUSTER to tell you that path B has been reestablished. |
| **5** | Remove path A. |
| **6** | After the poller has discovered that path A is faulty, reconnect path A. |
| **7** | Wait two poller intervals[1] to make sure that the poller has reestablished path A. |

[1]Approximately 10 seconds at the default system parameter settings

If both paths are lost at the same time, the virtual circuits are lost between the port with the broken cables and all other ports in the cluster. This condition will in turn result in loss of SCS connections over the broken virtual circuits. However, recovery from this situation is automatic after an interruption in service on the affected computer. The length of the interruption varies, but it is approximately two poller intervals at the default system parameter settings.

### C.10.8 Verifying LAN Connections

The Local Area OpenVMS Cluster Network Failure Analysis Program described in Section D.4 uses the HELLO datagram messages to verify continuously the network paths (channels) used by PEDRIVER. This verification process, combined with physical description of the network, can:

- Isolate failing network components
- Group failing channels together and map them onto the physical network description
- Call out the common components related to the channel failures

## C.11 Analyzing Error-Log Entries for Port Devices

Monitoring events recorded in the error log can help you anticipate and avoid potential problems. From the total error count (displayed by the DCL command SHOW DEVICES *device-name*), you can determine whether errors are increasing. If so, you should examine the error log.

### C.11.1 Examine the Error Log

The DCL command ANALYZE/ERROR_LOG invokes the Error Log utility to report the contents of an error-log file.

**Reference:** For more information about the Error Log utility, see the *OpenVMS System Management Utilities Reference Manual*.

Some error-log entries are informational only while others require action.

**Table C–5 Informational and Other Error-Log Entries**

| Error Type | Action Required? | Purpose |
|---|---|---|
| *Informational* error-log entries require no action. For example, if you shut down a computer in the cluster, all other active computers that have open virtual circuits between themselves and the computer that has been shut down make entries in their error logs. Such computers record up to three errors for the event:<br>• Path A received no response.<br>• Path B received no response.<br>• The virtual circuit is being closed. | No | These messages are normal and reflect the change of state in the circuits to the computer that has been shut down. |
| *Other* error-log entries indicate problems that degrade operation or nonfatal hardware problems. The operating system might continue to run satisfactorily under these conditions. | Yes | Detecting these problems early is important to preventing nonfatal problems (such as loss of a single CI path) from becoming serious problems (such as loss of both paths). |

## C.11.2 Formats

Errors and other events on the CI, DSSI, or LAN cause port drivers to enter information in the system error log in one of two formats:

- Device attention

  Device-attention entries for the CI record events that, in general, are indicated by the setting of a bit in a hardware register. For the LAN, device-attention entries typically record errors on a LAN adapter device.

- Logged message

  Logged-message entries record the receipt of a message packet that contains erroneous data or that signals an error condition.

Sections C.11.3 and C.11.6 describe those formats.

## C.11.3 CI Device-Attention Entries

Example C–6 shows device-attention entries for the CI. The left column gives the name of a device register or a memory location. The center column gives the value contained in that register or location, and the right column gives an interpretation of that value.

**Example C–6 CI Device-Attention Entries**

```
************************ ENTRY    83. **************************** ❶
ERROR SEQUENCE 10.                    LOGGED ON:      SID 0150400A
DATE/TIME 15-JAN-1994 11:45:27.61                 SYS_TYPE 01010000 ❷
DEVICE ATTENTION    KA780                                           ❸
                    SCS NODE: MARS

CI SUB-SYSTEM, MARS$PAA0: - PORT POWER DOWN                         ❹

      CNFGR           00800038
                                    ADAPTER IS CI
                                    ADAPTER POWER-DOWN
      PMCSR           000000CE
                                    MAINTENANCE TIMER DISABLE
                                    MAINTENANCE INTERRUPT ENABLE
                                    MAINTENANCE INTERRUPT FLAG
                                    PROGRAMMABLE STARTING ADDRESS
                                    UNINITIALIZED STATE
      PSR             80000001
                                    RESPONSE QUEUE AVAILABLE
                                    MAINTENANCE ERROR
      PFAR            00000000
      PESR            00000000
      PPR             03F80001
      UCB$B_ERTCNT          32                                        ❺
                                    50. RETRIES REMAINING
      UCB$B_ERTMAX          32                                        ❻
                                    50. RETRIES ALLOWABLE
      UCB$L_CHAR      0C450000
                                    SHAREABLE
                                    AVAILABLE
                                    ERROR LOGGING
                                    CAPABLE OF INPUT
                                    CAPABLE OF OUTPUT
      UCB$W_STS           0010
                                    ONLINE
      UCB$W_ERRCNT        000B                                        ❼
                                    11. ERRORS THIS UNIT
```

The following table describes the device-attention entries in Example C–6.

| Entry | Description |
|---|---|
| ❶ | The first two lines are the entry heading. These lines contain the number of the entry in this error log file, the sequence number of this error, and the identification number (SID) of this computer. Each entry in the log file contains such a heading. |
| ❷ | This line contains the date, the time, and the computer type. |

| Entry | Description |
|---|---|
| ❸ | The next two lines contain the entry type, the processor type (KA780), and the computer's SCS node name. |
| ❹ | This line shows the name of the subsystem and the device that caused the entry and the reason for the entry. The CI subsystem's device PAA0 on MARS was powered down.<br>The next 15 lines contain the names of hardware registers in the port, their contents, and interpretations of those contents. See the appropriate CI hardware manual for a description of all the CI port registers. |
| ❺ | The UCB$B_ERTCNT field contains the number of reinitializations that the port driver can still attempt. The difference between this value and UCB$B_ERTMAX is the number of reinitializations already attempted. |
| ❻ | The UCB$B_ERTMAX field contains the maximum number of times the port can be reinitialized by the port driver. |
| ❼ | The UCB$W_ERRCNT field contains the total number of errors that have occurred on this port since it was booted. This total includes both errors that caused reinitialization of the port and errors that did not. |

## C.11.4 Error Recovery

The CI port can recover from many errors, but not all. When an error occurs from which the CI cannot recover, the following process occurs:

| Step | Action |
|---|---|
| **1** | The port notifies the port driver. |
| **2** | The port driver logs the error and attempts to reinitialize the port. |
| **3** | If the port fails after 50 such initialization attempts, the driver takes it off line, unless the system disk is connected to the failing port or unless this computer is supposed to be a cluster member. |
| **4** | If the CI port is required for system disk access or cluster participation and all 50 reinitialization attempts have been used, then the computer bugchecks with a CIPORT-type bugcheck. |

Once a CI port is off line, you can put the port back on line only by rebooting the computer.

## C.11.5 LAN Device-Attention Entries

Example C–7 shows device-attention entries for the LAN. The left column gives the name of a device register or a memory location. The center column gives the value contained in that register or location, and the right column gives an interpretation of that value.

### Example C–7 LAN Device-Attention Entry

```
************************ ENTRY   80. ****************************  ❶
ERROR SEQUENCE 26.                   LOGGED ON:      SID 08000000
DATE/TIME 15-JAN-1994 11:30:53.07               SYS_TYPE 01010000  ❷
DEVICE ATTENTION  KA630                                            ❸
                  SCS NODE: PHOBOS
NI-SCS SUB-SYSTEM, PHOBOS$PEA0:                                    ❹
       FATAL ERROR DETECTED BY DATALINK                            ❺
```

(continued on next page)

**Example C–7 (Cont.) LAN Device-Attention Entry**

```
STATUS1          0000002C                                    ❻
STATUS2          00000000
DATALINK UNIT       0001                                     ❼
DATALINK NAME    41515803                                    ❽
                 00000000
                 00000000
                 00000000
                                 DATALINK NAME = XQA1:
REMOTE NODE      00000000                                    ❾
                 00000000
                 00000000
                 00000000
REMOTE ADDR      00000000                                    ❿
                     0000
LOCAL ADDR       000400AA                                    ⓫
                     4C07
                                 ETHERNET ADDR = AA-00-04-00-07-4C
ERROR CNT            0001                                    ⓬
                                 1. ERROR OCCURRENCES THIS ENTRY
UCB$W_ERRCNT         0007
                                 7. ERRORS THIS UNIT
```

The following table describes the LAN device-attention entries in Example C–7.

| Entry | Description |
|---|---|
| ❶ | The first two lines are the entry heading. These lines contain the number of the entry in this error log file, the sequence number of this error, and the identification number (SID) of this computer. Each entry in the log file contains such a heading. |
| ❷ | This line contains the date and time and the computer type. |
| ❸ | The next two lines contain the entry type, the processor type (KA630), and the computer's SCS node name. |
| ❹ | This line shows the name of the subsystem and component that caused the entry. |
| ❺ | This line shows the reason for the entry. The LAN driver has shut down the data link because of a fatal error. The data link will be restarted automatically, if possible. |
| ❻ | STATUS1 shows the I/O completion status returned by the LAN driver. STATUS2 is the VCI event code delivered to PEDRIVER by the LAN driver. The event values and meanings are described in the following table: |

| Event Code | Meaning |
|---|---|
| 1200 | Port usable |
| 1201 | Port unusable |
| 1202 | Change address |

If a message transmit was involved, the status applies to that transmit.

| | |
|---|---|
| ❼ | DATALINK UNIT shows the unit number of the LAN device on which the error occurred. |
| ❽ | DATALINK NAME is the name of the LAN device on which the error occurred. |
| ❾ | REMOTE NODE is the name of the remote node to which the packet was being sent. If zeros are displayed, either no remote node was available or no packet was associated with the error. |
| ❿ | REMOTE ADDR is the LAN address of the remote node to which the packet was being sent. If zeros are displayed, no packet was associated with the error. |
| ⓫ | LOCAL ADDR is the LAN address of the local node. |

| Entry | Description |
|---|---|
| ⓬ | ERROR CNT. Because some errors can occur at extremely high rates, some error log entries represent more than one occurrence of an error. This field indicates how many. The errors counted occurred in the 3 seconds preceding the timestamp on the entry. |

## C.11.6 Logged Message Entries

Logged-message entries are made when the CI or LAN port receives a response that contains either data that the port driver cannot interpret or an error code in status field of the response.

Example C–8 shows a logged-message entry with an error code in the status field PPD$B_STATUS for a CI port.

### Example C–8 CI Port Logged-Message Entry

```
************************ ENTRY     3. **************************** ❶
ERROR SEQUENCE 3.                              LOGGED ON SID 01188542

ERL$LOGMESSAGE, 15-JAN-1994 13:40:25.13                              ❷
               KA780 REV #3. SERIAL #1346.    MFG PLANT 15.          ❸

CI SUB-SYSTEM, MARS$PAA0:                                            ❹
DATA CABLE(S) STATE CHANGE - PATH #0. WENT FROM GOOD TO BAD          ❺

      LOCAL STATION ADDRESS, 000000000002 (HEX)                      ❻

      LOCAL SYSTEM ID, 000000000001 (HEX)                            ❼

      REMOTE STATION ADDRESS, 000000000004 (HEX)                     ❽

      REMOTE SYSTEM ID, 0000000000A9 (HEX)                           ❾

      UCB$B_ERTCNT          32                                       ❿
                                    50. RETRIES REMAINING
      UCB$B_ERTMAX          32
                                    50. RETRIES ALLOWABLE
      UCB$W_ERRCNT        0001
                                    1. ERRORS THIS UNIT
      PPD$B_PORT            04                                       ⓫
                                    REMOTE NODE #4.
      PPD$B_STATUS          A5                                       ⓬
                                    FAIL
                                    PATH #0., NO RESPONSE
                                    PATH #1., "ACK" OR NOT USED
                                    NO PATH
      PPD$B_OPC             05                                       ⓭
                                    IDREQ
      PPD$B_FLAGS           03                                       ⓮
                                    RESPONSE QUEUE BIT
                                    SELECT PATH #0.
```

(continued on next page)

**Example C–8 (Cont.) CI Port Logged-Message Entry**

```
"CI" MESSAGE                                                    ⓯

                00000000
                00000000
                80000004
                0000FE15
                4F503000
                00000507
                00000000
                00000000
                00000000
                00000000
                00000000
                00000000
                00000000
                00000000
                00000000
                00000000
                00000000
```

The following table describes the logged-message entries in Example C–8.

| Entry | Description |
|---|---|
| ❶ | The first two lines are the entry heading. These lines contain the number of the entry in this error log file, the sequence number of the error, and the identification number (SID) of the computer. Each entry in the log file contains a heading. |
| ❷ | This line contains the entry type, the date, and the time. |
| ❸ | This line contains the processor type (KA780), the hardware revision number of the computer (REV #3), the serial number of the computer (SERIAL #1346), and the plant number ( 15 ). |
| ❹ | This line shows the name of the subsystem and the device that caused the entry. |
| ❺ | This line gives the reason for the entry (one or more data cables have changed state) and a more detailed reason for the entry. Path 0, which the port used successfully before, cannot be used now.<br>**Note:** ANALYZE/ERROR_LOG uses the notation “path 0” and “path 1”; cable labels use the notation “path A ( =0 )” and “path B ( =1 )”. |
| ❻ | The local (❻) and remote (❽) station addresses are the port numbers (range 0 to 15) of the local and remote ports. The port numbers are set in hardware switches by Compaq support representatives. |
| ❼ | The local (❼) and remote (❾) system IDs are the SCS system IDs set by the system parameter SCSSYSTEMID for the local and remote systems. For HSC subsystems, the system ID is set with the HSC console. |
| ❽ | See ❻. |
| ❾ | See ❼. |
| ❿ | The next three lines consist of the entry fields that begin with UCB$. These fields give information on the contents of the unit control block (UCB) for this CI device. |
| ⓫ | The lines that begin with PPD$ are fields in the message packet that the local port has received. PPD$B_PORT contains the station address of the remote port. In a loopback datagram, however, this field contains the local station address. |
| ⓬ | The PPD$B_STATUS field contains information about the nature of the failure that occurred during the current operation. When the operation completes without error, ERF prints the word NORMAL beside this field; otherwise, ERF decodes the error information contained in PPD$B_STATUS. Here a NO PATH error occurred because of a lack of response on path 0, the selected path. |

| Entry | Description |
|---|---|
| ⓭ | The PPD$B_OPC field contains the code for the operation that the port was attempting when the error occurred. The port was trying to send a request-for-ID message. |
| ⓮ | The PPD$B_FLAGS field contains bits that indicate, among other things, the path that was selected for the operation. |
| ⓯ | "CI" MESSAGE is a hexadecimal listing of bytes 16 through 83 (decimal) of the response (message or datagram). Because responses are of variable length, depending on the port opcode, bytes 16 through 83 may contain either more or fewer bytes than actually belong to the message. |

## C.11.7 Error-Log Entry Descriptions

This section describes error-log entries for the CI and LAN ports. Each entry shown is followed by a brief description of what the associated port driver (for example, PADRIVER, PBDRIVER, PEDRIVER) does, and the suggested action a system manager should take. In cases where you are advised to contact your Compaq support representative. and save crash dumps, it is important to capture the crash dumps as soon as possible after the error. For CI entries, note that path A and path 0 are the same path, and that path B and path 1 are the same path.

Table C–6 lists error-log messages.

**Table C–6 Port Messages for All Devices**

| Message | Result | User Action |
|---|---|---|
| BIIC FAILURE | The port driver attempts to reinitialize the port; after 50 failed attempts, it marks the device off line. | Contact your Compaq support representative. |
| CI PORT TIMEOUT | The port driver attempts to reinitialize the port; after 50 failed attempts, it marks the device off line. | Increase the PAPOLLINTERVAL system parameter. If the problem disappears and you are not running privileged user-written software, contact your Compaq support representative. |
| 11/750 CPU MICROCODE NOT ADEQUATE FOR PORT | The port driver sets the port off line with no retries attempted. In addition, if this port is needed because the computer is booted from an HSC subsystem or is participating in a cluster, the computer bugchecks with a UCODEREV code bugcheck. | Read the appropriate section in the current OpenVMS Cluster Software SPD for information on required computer microcode revisions. Contact your Compaq support representative, if necessary. |
| PORT MICROCODE REV NOT CURRENT, BUT SUPPORTED | The port driver detected that the microcode is not at the current level, but the port driver will continue normally. This error is logged as a warning only. | Contact your Compaq support representative when it is convenient to have the microcode updated. |
| PORT MICROCODE REV NOT SUPPORTED | The port driver sets the port off line without attempting any retries. | Read the OpenVMS Cluster Software SPD for information on the required CI port microcode revisions. Contact your Compaq support representative, if necessary. |

(continued on next page)

**Table C–6 (Cont.) Port Messages for All Devices**

| Message | Result | User Action |
|---|---|---|
| DATA CABLE(S) STATE CHANGE\ CABLES HAVE GONE FROM CROSSED TO UNCROSSED | The port driver logs this event. | No action needed. |
| DATA CABLE(S) STATE CHANGE\ CABLES HAVE GONE FROM UNCROSSED TO CROSSED | The port driver logs this event. | Check for crossed cable pairs. (See Section C.10.5.) |
| DATA CABLE(S) STATE CHANGE\ PATH 0. WENT FROM BAD TO GOOD | The port driver logs this event. | No action needed. |
| DATA CABLE(S) STATE CHANGE\ PATH 0. WENT FROM GOOD TO BAD | The port driver logs this event. | Check path A cables to see that they are not broken or improperly connected. |
| DATA CABLE(S) STATE CHANGE\ PATH 0. LOOPBACK IS NOW GOOD, UNCROSSED | The port driver logs this event. | No action needed. |
| DATA CABLE(S) STATE CHANGE\ PATH 0. LOOPBACK WENT FROM GOOD TO BAD | The port driver logs this event. | Check for crossed cable pairs or faulty CI hardware. (See Sections C.10.3 and C.10.5.) |
| DATA CABLE(S) STATE CHANGE\ PATH 1. WENT FROM BAD TO GOOD | The port driver logs this event. | No action needed. |
| DATA CABLE(S) STATE CHANGE\ PATH 1. WENT FROM GOOD TO BAD | The port driver logs this event. | Check path B cables to see that they are not broken or improperly connected. |
| DATA CABLE(S) STATE CHANGE\ PATH 1. LOOPBACK IS NOW GOOD, UNCROSSED | The port driver logs this event. | No action needed. |
| DATA CABLE(S) STATE CHANGE\ PATH 1. LOOPBACK WENT FROM GOOD TO BAD | The port driver logs this event. | Check for crossed cable pairs or faulty CI hardware. (See Sections C.10.3 and C.10.5.) |
| DATAGRAM FREE QUEUE INSERT FAILURE | The port driver attempts to reinitialize the port; after 50 failed attempts, it marks the device off line. | Contact your Compaq support representative. This error is caused by a failure to obtain access to an interlocked queue. Possible sources of the problem are CI hardware failures, or memory, SBI (11/780), CMI (11/750), or BI (8200, 8300, and 8800) contention. |
| DATAGRAM FREE QUEUE REMOVE FAILURE | The port driver attempts to reinitialize the port; after 50 failed attempts, it marks the device off line. | Contact your Compaq support representative. This error is caused by a failure to obtain access to an interlocked queue. Possible sources of the problem are CI hardware failures, or memory, SBI (11/780), CMI (11/750), or BI (8200, 8300, and 8800) contention. |

(continued on next page)

**Table C–6 (Cont.) Port Messages for All Devices**

| Message | Result | User Action |
|---|---|---|
| FAILED TO LOCATE PORT MICROCODE IMAGE | The port driver marks the device off line and makes no retries. | Make sure console volume contains the microcode file CI780.BIN (for the CI780, CI750, or CIBCI) or the microcode file CIBCA.BIN for the CIBCA–AA. Then reboot the computer. |
| HIGH PRIORITY COMMAND QUEUE INSERT FAILURE | The port driver attempts to reinitialize the port; after 50 failed attempts, it marks the device off line. | Contact your Compaq support representative. This error is caused by a failure to obtain access to an interlocked queue. Possible sources of the problem are CI hardware failures, or memory, SBI (11/780), CMI (11/750), or BI (8200, 8300, and 8800) contention. |
| MSCP ERROR LOGGING DATAGRAM RECEIVED | On receipt of an error message from the HSC subsystem, the port driver logs the error and takes no other action. You should disable the sending of HSC informational error-log datagrams with the appropriate HSC console command because such datagrams take considerable space in the error-log data file. | Error-log datagrams are useful to read only if they are not captured on the HSC console for some reason (for example, if the HSC console ran out of paper.) This logged information duplicates messages logged on the HSC console. |
| INAPPROPRIATE SCA CONTROL MESSAGE | The port driver closes the port-to-port virtual circuit to the remote port. | Contact your Compaq support representative. Save the error logs and the crash dumps from the local and remote computers. |
| INSUFFICIENT NON-PAGED POOL FOR INITIALIZATION | The port driver marks the device off line and makes no retries. | Reboot the computer with a larger value for NPAGEDYN or NPAGEVIR. |
| LOW PRIORITY CMD QUEUE INSERT FAILURE | The port driver attempts to reinitialize the port; after 50 failed attempts, it marks the device off line. | Contact your Compaq support representative. This error is caused by a failure to obtain access to an interlocked queue. Possible sources of the problem are CI hardware failures, or memory, SBI (11/780), CMI (11/750), or BI (8200, 8300, and 8800) contention. |
| MESSAGE FREE QUEUE INSERT FAILURE | The port driver attempts to reinitialize the port; after 50 failed attempts, it marks the device off line. | Contact your Compaq support representative. This error is caused by a failure to obtain access to an interlocked queue. Possible sources of the problem are CI hardware failures, or memory, SBI (11/780), CMI (11/750), or BI (8200, 8300, and 8800) contention. |
| MESSAGE FREE QUEUE REMOVE FAILURE | The port driver attempts to reinitialize the port; after 50 failed attempts, it marks the device off line. | Contact your Compaq support representative. This error is caused by a failure to obtain access to an interlocked queue. Possible sources of the problem are CI hardware failures, or memory, SBI (11/780), CMI (11/750), or BI (8200, 8300, and 8800) contention. |

(continued on next page)

**Table C–6 (Cont.) Port Messages for All Devices**

| Message | Result | User Action |
|---|---|---|
| MICRO-CODE VERIFICATION ERROR | The port driver detected an error while reading the microcode that it just loaded into the port. The driver attempts to reinitialize the port; after 50 failed attempts, it marks the device off line. | Contact your Compaq support representative. |
| NO PATH-BLOCK DURING VIRTUAL CIRCUIT CLOSE | The port driver attempts to reinitialize the port; after 50 failed attempts, it marks the device off line. | Contact your Compaq support representative. Save the error log and a crash dump from the local computer. |
| NO TRANSITION FROM UNINITIALIZED TO DISABLED | The port driver attempts to reinitialize the port; after 50 failed attempts, it marks the device off line. | Contact your Compaq support representative. |
| PORT ERROR BIT(S) SET | The port driver attempts to reinitialize the port; after 50 failed attempts, it marks the device off line. | A **maintenance timer expiration** bit may mean that the PASTIMOUT system parameter is set too low and should be increased, especially if the local computer is running privileged user-written software. For all other bits, call your Compaq support representative. |
| PORT HAS CLOSED VIRTUAL CIRCUIT | The port driver closed the virtual circuit that the local port opened to the remote port. | Check the PPD$B_STATUS field of the error-log entry for the reason the virtual circuit was closed. This error is normal if the remote computer failed or was shut down. For PEDRIVER, ignore the PPD$B_OPC field value; it is an unknown opcode.<br>If PEDRIVER logs a large number of these errors, there may be a problem either with the LAN or with a remote system, or nonpaged pool may be insufficient on the local system. |
| PORT POWER DOWN | The port driver halts port operations and then waits for power to return to the port hardware. | Restore power to the port hardware. |
| PORT POWER UP | The port driver reinitializes the port and restarts port operations. | No action needed. |
| RECEIVED CONNECT WITHOUT PATH-BLOCK | The port driver attempts to reinitialize the port; after 50 failed attempts, it marks the device off line. | Contact your Compaq support representative. Save the error log and a crash dump from the local computer. |

(continued on next page)

**Table C–6 (Cont.) Port Messages for All Devices**

| Message | Result | User Action |
|---|---|---|
| REMOTE SYSTEM CONFLICTS WITH KNOWN SYSTEM | The configuration poller discovered a remote computer with SCSSYSTEMID and/or SCSNODE equal to that of another computer to which a virtual circuit is already open. | Shut down the new computer as soon as possible. Reboot it with a unique SCSYSTEMID and SCSNODE. Do not leave the new computer up any longer than necessary. If you are running a cluster, and two computers with conflicting identity are polling when any other virtual circuit failure takes place in the cluster, then computers in the cluster may shut down with a CLUEXIT bugcheck. |
| RESPONSE QUEUE REMOVE FAILURE | The port driver attempts to reinitialize the port; after 50 failed attempts, it marks the device off line. | Contact your Compaq support representative. This error is caused by a failure to obtain access to an interlocked queue. Possible sources of the problem are CI hardware failures, or memory, SBI (11/780), CMI (11/750), or BI (8200, 8300, and 8800) contention. |
| SCSSYSTEMID MUST BE SET TO NON-ZERO VALUE | The port driver sets the port off line without attempting any retries. | Reboot the computer with a conversational boot and set the SCSSYSTEMID to the correct value. At the same time, check that SCSNODE has been set to the correct nonblank value. |
| SOFTWARE IS CLOSING VIRTUAL CIRCUIT | The port driver closes the virtual circuit to the remote port. | Check error-log entries for the cause of the virtual circuit closure. Faulty transmission or reception on both paths, for example, causes this error and may be detected from the one or two previous error-log entries noting bad paths to this remote computer. |
| SOFTWARE SHUTTING DOWN PORT | The port driver attempts to reinitialize the port; after 50 failed attempts, it marks the device off line. | Check other error-log entries for the possible cause of the port reinitialization failure. |
| UNEXPECTED INTERRUPT | The port driver attempts to reinitialize the port; after 50 failed attempts, it marks the device off line. | Contact your Compaq support representative. |
| UNRECOGNIZED SCA PACKET | The port driver closes the virtual circuit to the remote port. If the virtual circuit is already closed, the port driver inhibits datagram reception from the remote port. | Contact your Compaq support representative. Save the error-log file that contains this entry and the crash dumps from both the local and remote computers. |
| VIRTUAL CIRCUIT TIMEOUT | The port driver closes the virtual circuit that the local CI port opened to the remote port. This closure occurs if the remote computer is running CI microcode Version 7 or later, and if the remote computer has failed to respond to any messages sent by the local computer. | This error is normal if the remote computer has halted, failed, or was shut down. This error may mean that the local computer's TIMVCFAIL system parameter is set too low, especially if the remote computer is running privileged user-written software. |

(continued on next page)

**Table C–6 (Cont.) Port Messages for All Devices**

| Message | Result | User Action |
|---|---|---|
| INSUFFICIENT NON-PAGED POOL FOR VIRTUAL CIRCUITS | The port driver closes virtual circuits because of insufficient pool. | Enter the DCL command SHOW MEMORY to determine pool requirements, and then adjust the appropriate system parameter requirements. |

The descriptions in Table C–7 apply only to LAN devices.

**Table C–7 Port Messages for LAN Devices**

| Message | Completion Status | Explanation | User Action |
|---|---|---|---|
| FATAL ERROR DETECTED BY DATALINK | First longword SS$_ NORMAL (00000001), second longword (00001201) | The LAN driver stopped the local area OpenVMS Cluster protocol on the device. This completion status is returned when the SYS$LAVC_STOP_BUS routine completes successfully. The SYS$LAVC_STOP_BUS routine is called either from within the LAVC$STOP_BUS.MAR program found in SYS$EXAMPLES or from a user-written program. The local area OpenVMS Cluster protocol remains stopped on the specified device until the SYS$LAVC_START_BUS routine executes successfully. The SYS$LAVC_ START_BUS routine is called from within the LAVC$START_BUS.MAR program found in SYS$EXAMPLES or from a user-written program. | If the protocol on the device was stopped inadvertently, then restart the protocol by assembling and executing the LAVC$START_BUS program found in SYS$EXAMPLES.<br>**Reference:** See Appendix D for an explanation of the local area OpenVMS Cluster sample programs. Otherwise, this error message can be safely ignored. |
| | First longword is any value other than (00000001), second longword (00001201) | The LAN driver has shut down the device because of a fatal error and is returning all outstanding transmits with SS$_OPINCOMPL. The LAN device is restarted automatically. | Infrequent occurrences of this error are typically not a problem. If the error occurs frequently or is accompanied by loss or reestablishment of connections to remote computers, there may be a hardware problem. Check for the proper LAN adapter revision level or contact your Compaq support representative. |
| | First longword (undefined), second longword (00001200) | The LAN driver has restarted the device successfully after a fatal error. This error-log message is usually preceded by a FATAL ERROR DETECTED BY DATALINK error-log message whose first completion status longword is anything other than 00000001 and whose second completion status longword is 00001201. | No action needed. |
| TRANSMIT ERROR FROM DATALINK | SS$_OPINCOMPL (000002D4) | The LAN driver is in the process of restarting the data link because an error forced the driver to shut down the controller and all users (see FATAL ERROR DETECTED BY DATALINK). | |

(continued on next page)

**Table C–7 (Cont.) Port Messages for LAN Devices**

| Message | Completion Status | Explanation | User Action |
|---|---|---|---|
| | SS$_DEVREQERR (00000334) | The LAN controller tried to transmit the packet 16 times and failed because of defers and collisions. This condition indicates that LAN traffic is heavy. | |
| | SS$_DISCONNECT (0000204C) | There was a loss of carrier during or after the transmit. | The port emulator automatically recovers from any of these errors, but many such errors indicate either that the LAN controller is faulty or that the LAN is overloaded. If you suspect either of these conditions, contact your Compaq support representative. |
| INVALID CLUSTER PASSWORD RECEIVED | | A computer is trying to join the cluster using the correct cluster group number for this cluster but an invalid password. The port emulator discards the message. The probable cause is that another cluster on the LAN is using the same cluster group number. | Provide all clusters on the same LAN with unique cluster group numbers. |
| NISCS PROTOCOL VERSION MISMATCH RECEIVED | | A computer is trying to join the cluster using a version of the cluster LAN protocol that is incompatible with the one in use on this cluster. | Install a version of the operating system that uses a compatible protocol, or change the cluster group number so that the computer joins a different cluster. |

## C.12 OPA0 Error-Message Logging and Broadcasting

Port drivers detect certain error conditions and attempt to log them. The port driver attempts both OPA0 error broadcasting and standard error logging under any of the following circumstances:

- The system disk has not yet been mounted.
- The system disk is undergoing mount verification.
- During mount verification, the system disk drive contains the wrong volume.
- Mount verification for the system disk has timed out.
- The local computer is participating in a cluster, and quorum has been lost.

Note the implicit assumption that the system and error-logging devices are one and the same.

The following table describes error-logging methods and their reliability.

| Method | Reliability | Comments |
|---|---|---|
| Standard error logging to an error-logging device. | Under some circumstances, attempts to log errors to the error-logging device can fail. Such failures can occur because the error-logging device is not accessible when attempts are made to log the error condition. | Because of the central role that the port device plays in clusters, the loss of error-logged information in such cases makes it difficult to diagnose and fix problems. |
| Broadcasting selected information about the error condition to OPA0. (This is in addition to the port driver's attempt to log the error condition to the error-logging device.) | This method of reporting errors is not entirely reliable, because some error conditions may not be reported due to the way OPA0 error broadcasting is performed. This situation occurs whenever a second error condition is detected before the port driver has been able to broadcast the first error condition to OPA0. In such a case, only the first error condition is reported to OPA0, because that condition is deemed to be the more important one. | This second, redundant method of error logging captures at least some of the information about port-device error conditions that would otherwise be lost. |

**Note:** Certain error conditions are always broadcast to OPA0, regardless of whether the error-logging device is accessible. In general, these are errors that cause the port to shut down either permanently or temporarily.

## C.12.1 OPA0 Error Messages

One OPA0 error message for each error condition is always logged. The text of each error message is similar to the text in the summary displayed by formatting the corresponding standard error-log entry using the Error Log utility. (See Section C.11.7 for a list of Error Log utility summary messages and their explanations.)

Table C–8 lists the OPA0 error messages. The table is divided into units by error type. Many of the OPA0 error messages contain some optional information, such as the remote port number, CI packet information (flags, port operation code, response status, and port number fields), or specific CI port registers. The codes specify whether the message is always logged on OPA0 or is logged only when the system device is inaccessible.

**Table C–8 OPA0 Messages**

| Error Message | Logged or Inaccessible |
|---|---|
| **Software Errors During Initialization** | |
| %Pxxn, Insufficient Non-Paged Pool for Initialization | Logged |
| %Pxxn, Failed to Locate Port Micro-code Image | Logged |
| %Pxxn, SCSSYSTEMID has NOT been set to a Non-Zero Value | Logged |

(continued on next page)

**Table C–8 (Cont.) OPA0 Messages**

| Hardware Errors | |
|---|---|
| %Pxxn, BIIC failure—BICSR/BER/CNF xxxxxx/xxxxxx/xxxxxx | Logged |
| %Pxxn, Micro-code Verification Error | Logged |
| %Pxxn, Port Transition Failure—CNF/PMC/PSR xxxxxx/xxxxxx/xxxxxx | Logged |
| %Pxxn, Port Error Bit(s) Set—CNF/PMC/PSR xxxxxx/xxxxxx/xxxxxx | Logged |
| %Pxxn, Port Power Down | Logged |
| %Pxxn, Port Power Up | Logged |
| %Pxxn, Unexpected Interrupt—CNF/PMC/PSR xxxxxx/xxxxxx/xxxxxx | Logged |
| %Pxxn, CI Port Timeout | Logged |
| %Pxxn, CI port ucode not at required rev level. —RAM/PROM rev is xxxx/xxxx | Logged |
| %Pxxn, CI port ucode not at current rev level.—RAM/PROM rev is xxxx/xxxx | Logged |
| %Pxxn, CPU ucode not at required rev level for CI activity | Logged |
| **Queue Interlock Failures** | |
| %Pxxn, Message Free Queue Remove Failure | Logged |
| %Pxxn, Datagram Free Queue Remove Failure | Logged |
| %Pxxn, Response Queue Remove Failure | Logged |
| %Pxxn, High Priority Command Queue Insert Failure | Logged |
| %Pxxn, Low Priority Command Queue Insert Failure | Logged |
| %Pxxn, Message Free Queue Insert Failure | Logged |
| %Pxxn, Datagram Free Queue Insert Failure | Logged |
| **Errors Signaled with a CI Packet** | |
| %Pxxn, Unrecognized SCA Packet—FLAGS/OPC/STATUS/PORT xx/xx/xx/xx | Logged |
| %Pxxn, Port has Closed Virtual Circuit—REMOTE PORT[1] xxx | Logged |
| %Pxxn, Software Shutting Down Port | Logged |
| %Pxxn, Software is Closing Virtual Circuit—REMOTE PORT[1] xxx | Logged |
| %Pxxn, Received Connect Without Path-Block—FLAGS/OPC/STATUS/PORT xx/xx/xx/xx | Logged |
| %Pxxn, Inappropriate SCA Control Message—FLAGS/OPC/STATUS/PORT xx/xx/xx/xx | Logged |
| %Pxxn, No Path-Block During Virtual Circuit Close—REMOTE PORT[1] xxx | Logged |
| %Pxxn, HSC Error Logging Datagram Received Inaccessible—REMOTE PORT[1] xxx | Inaccessible |
| %Pxxn, Remote System Conflicts with Known System—REMOTE PORT[1] xxx | Logged |
| %Pxxn, Virtual Circuit Timeout—REMOTE PORT[1] xxx | Logged |
| %Pxxn, Parallel Path is Closing Virtual Circuit— REMOTE PORT[1] xxx | Logged |

[1]If the port driver can identify the remote SCS node name of the affected computer, the driver replaces the "REMOTE PORT *xxx*" text with "REMOTE SYSTEM X...", where X... is the value of the system parameter SCSNODE on the remote computer. If the remote SCS node name is not available, the port driver uses the existing message format.

**Key to CI Port Registers:**

CNF—configuration register
PMC—port maintenance and control register
PSR—port status register

See also the CI hardware documentation for a detailed description of the CI port registers.

(continued on next page)

**Table C–8 (Cont.) OPA0 Messages**

| Error Message | Logged or Inaccessible |
|---|---|
| **Errors Signaled with a CI Packet** | |
| %Pxxn, Insufficient Nonpaged Pool for Virtual Circuits | Logged |
| **Cable Change-of-State Notification** | |
| %Pxxn, Path #0. Has gone from GOOD to BAD—REMOTE PORT[1] xxx | Inaccessible |
| %Pxxn, Path #1. Has gone from GOOD to BAD—REMOTE PORT[1] xxx | Inaccessible |
| %Pxxn, Path #0. Has gone from BAD to GOOD—REMOTE PORT[1] xxx | Inaccessible |
| %Pxxn, Path #1. Has gone from BAD to GOOD—REMOTE PORT[1] xxx | Inaccessible |
| %Pxxn, Cables have gone from UNCROSSED to CROSSED—REMOTE PORT[1] xxx | Inaccessible |
| %Pxxn, Cables have gone from CROSSED to UNCROSSED—REMOTE PORT[1] xxx | Inaccessible |
| %Pxxn, Path #0. Loopback has gone from GOOD to BAD—REMOTE PORT[1] xxx | Logged |
| %Pxxn, Path #1. Loopback has gone from GOOD to BAD—REMOTE PORT[1] xxx | Logged |
| %Pxxn, Path #0. Loopback has gone from BAD to GOOD—REMOTE PORT[1] xxx | Logged |
| %Pxxn, Path #1. Loopback has gone from BAD to GOOD—REMOTE PORT[1] xxx | Logged |
| %Pxxn, Path #0. Has become working but CROSSED to Path #1.— REMOTE PORT[1] xxx | Inaccessible |
| %Pxxn, Path #1. Has become working but CROSSED to Path #0.— REMOTE PORT[1] xxx | Inaccessible |

[1]If the port driver can identify the remote SCS node name of the affected computer, the driver replaces the "REMOTE PORT *xxx*" text with "REMOTE SYSTEM X...", where X... is the value of the system parameter SCSNODE on the remote computer. If the remote SCS node name is not available, the port driver uses the existing message format.

## C.12.2 CI Port Recovery

Two other messages concerning the CI port appear on OPA0:

```
%Pxxn, CI port is reinitializing (xxx retries left.)
%Pxxn, CI port is going off line.
```

The first message indicates that a previous error requiring the port to shut down is recoverable and that the port will be reinitialized. The "*xxx* retries left" specifies how many more reinitializations are allowed before the port must be left permanently off line. Each reinitialization of the port (for reasons other than power fail recovery) causes approximately 2 KB of nonpaged pool to be lost.

The second message indicates that a previous error is not recoverable and that the port will be left off line. In this case, the only way to recover the port is to reboot the computer.

# D

# Sample Programs for LAN Control

Sample programs are provided to start and stop the NISCA protocol on a LAN adapter, and to enable LAN network failure analysis. The following programs are located in SYS$EXAMPLES:

| Program | Description |
|---|---|
| LAVC$START_BUS.MAR | Starts the NISCA protocol on a specified LAN adapter. |
| LAVC$STOP_BUS.MAR | Stops the NISCA protocol on a specified LAN adapter. |
| LAVC$FAILURE_ANALYSIS.MAR | Enables LAN network failure analysis. |
| LAVC$BUILD.COM | Assembles and links the sample programs. |

**Reference:** The NISCA protocol, responsible for carrying messages across Ethernet and FDDI LANs to other nodes in the cluster, is described in Appendix F.

## D.1 Purpose of Programs

The port emulator driver, PEDRIVER, starts the NISCA protocol on all of the LAN adapters in the cluster. LAVC$START_BUS.MAR and LAVC$STOP_BUS.MAR are provided for cluster managers who want to split the network load according to protocol type and therefore do not want the NISCA protocol running on all of the LAN adapters.

**Reference:** See Section D.5 for information about editing and using the network failure analysis program.

## D.2 Starting the NISCA Protocol

The sample program LAVC$START_BUS.MAR, provided in SYS$EXAMPLES, starts the NISCA protocol on a specific LAN adapter.

To build the program, perform the following steps:

| Step | Action |
|---|---|
| **1** | Copy the files LAVC$START_BUS.MAR and LAVC$BUILD.COM from SYS$EXAMPLES to your local directory. |
| **2** | Assemble and link the sample program using the following command:<br>`$ @LAVC$BUILD.COM LAVC$START_BUS.MAR` |

### D.2.1 Start the Protocol

To start the protocol on a LAN adapter, perform the following steps:

| Step | Action |
|---|---|
| **1** | Use an account that has the PHY_IO privilege—you need this to run LAVC$START_BUS.EXE. |
| **2** | Define the foreign command (DCL symbol). |
| **3** | Execute the foreign command (LAVC$START_BUS.EXE), followed by the name of the LAN adapter on which you want to start the protocol. |

**Example:** The following example shows how to start the NISCA protocol on LAN adapter ETA0:

```
$ START_BUS:==$SYS$DISK:[ ]LAVC$START_BUS.EXE
$ START_BUS ETA
```

## D.3 Stopping the NISCA Protocol

The sample program LAVC$STOP_BUS.MAR, provided in SYS$EXAMPLES, stops the NISCA protocol on a specific LAN adapter.

**Caution:** Stopping the NISCA protocol on all LAN adapters causes satellites to hang and could cause cluster systems to fail with a CLUEXIT bugcheck.

Follow the steps below to build the program:

| Step | Action |
|---|---|
| **1** | Copy the files LAVC$STOP_BUS.MAR and LAVC$BUILD.COM from SYS$EXAMPLES to your local directory. |
| **2** | Assemble and link the sample program using the following command:<br>`$ @LAVC$BUILD.COM LAVC$STOP_BUS.MAR` |

### D.3.1 Stop the Protocol

To stop the NISCA protocol on a LAN adapter, perform the following steps:

| Step | Action |
|---|---|
| 1 | Use an account that has the PHY_IO privilege—you need this to run LAVC$STOP_BUS.EXE. |
| 2 | Define the foreign command (DCL symbol). |
| 3 | Execute the foreign command (LAVC$STOP_BUS.EXE), followed by the name of the LAN adapter on which you want to stop the protocol. |

**Example:** The following example shows how to stop the NISCA protocol on LAN adapter ETA0:

```
$ STOP_BUS:==$SYS$DISK[ ]LAVC$STOP_BUS.EXE
$ STOP_BUS ETA
```

### D.3.2 Verify Successful Execution

When the LAVC$STOP_BUS module executes successfully, the following device-attention entry is written to the system error log:

```
DEVICE ATTENTION . . .

NI-SCS SUB-SYSTEM . . .

FATAL ERROR DETECTED BY DATALINK . . .
```

In addition, the following hexidecimal values are written to the STATUS field of the entry:

First longword (00000001)
Second longword (00001201)

The error-log entry indicates expected behavior and can be ignored. However, if the first longword of the STATUS field contains a value other than hexidecimal value 00000001, an error has occurred and further investigation may be necessary.

## D.4 Analyzing Network Failures

LAVC$FAILURE_ANALYSIS.MAR is a sample program, located in SYS$EXAMPLES, that you can edit and use to help detect and isolate a failed network component. When the program executes, it provides the physical description of your cluster communications network to the set of routines that perform the failure analysis.

### D.4.1 Failure Analysis

Using the network failure analysis program can help reduce the time necessary for detection and isolation of a failing network component and, therefore, significantly increase cluster availability.

### D.4.2 How the LAVC$FAILURE_ANALYSIS Program Works

The following table describes how the LAVC$FAILURE_ANALYSIS program works.

| Step | Program Action |
|---|---|
| **1** | The program groups channels that fail and compares them with the physical description of the cluster network. |
| **2** | The program then develops a list of nonworking network components related to the failed channels and uses OPCOM messages to display the names of components with a probability of causing one or more channel failures.<br>If the network failure analysis cannot verify that a portion of a path (containing multiple components) works, the program:<br>1. Calls out the first component in the path as the primary suspect (%LAVC-W-PSUSPECT)<br>2. Lists the other components as secondary or additional suspects (%LAVC-I-ASUSPECT) |
| **3** | When the component works again, OPCOM displays the message %LAVC-S-WORKING. |

## D.5 Using the Network Failure Analysis Program

Table D–1 describes the steps you perform to edit and use the network failure analysis program.

**Table D–1 Procedure for Using the LAVC$FAILURE_ANALYSIS.MAR Program**

| Step | Action | Reference |
|---|---|---|
| **1** | Collect and record information specific to your cluster communications network. | Section D.5.1 |
| **2** | Edit a copy of LAVC$FAILURE_ANALYSIS.MAR to include the information you collected. | Section D.5.2 |
| **3** | Assemble, link, and debug the program. | Section D.5.3 |
| **4** | Modify startup files to run the program only on the node for which you supplied data. | Section D.5.4 |
| **5** | Execute the program on one or more of the nodes where you plan to perform the network failure analysis. | Section D.5.5 |
| **6** | Modify MODPARAMS.DAT to increase the values of nonpaged pool parameters. | Section D.5.6 |
| **7** | Test the Local Area OpenVMS Cluster Network Failure Analysis Program. | Section D.5.7 |

### D.5.1 Create a Network Diagram

Follow the steps in Table D–2 to create a physical description of the network configuration and include it in electronic form in the LAVC$FAILURE_ANALYSIS.MAR program.

**Table D–2 Creating a Physical Description of the Network**

| Step | Action | Comments |
|---|---|---|
| **1** | Draw a diagram of your OpenVMS Cluster communications network. | When you edit LAVC$FAILURE_ANALYSIS.MAR, you include this drawing (in electronic form) in the program. Your drawing should show the physical layout of the cluster and include the following components:<br>• LAN segments or rings<br>• LAN bridges<br>• Wiring concentrators, DELNI interconnects, or DEMPR repeaters<br>• LAN adapters<br>• VAX and Alpha systems<br>For large clusters, you may need to verify the configuration by tracing the cables. |
| **2** | Give each component in the drawing a unique label. | If your OpenVMS Cluster contains a large number of nodes, you may want to replace each node name with a shorter abbreviation. Abbreviating node names can help save space in the electronic form of the drawing when you include it in LAVC$FAILURE_ANALYSIS.MAR. For example, you can replace the node name ASTRA with A and call node ASTRA's two LAN adapters A1 and A2. |
| **3** | List the following information for each component:<br>• Unique label<br>• Type [SYSTEM, LAN_ADP, DELNI]<br>• Location (the physical location of the component)<br>• LAN address or addresses (if applicable) | Devices such as DELNI interconnects, DEMPR repeaters, and cables do not have LAN addresses. |

(continued on next page)

**Table D–2 (Cont.) Creating a Physical Description of the Network**

| Step | Action | Comments |
|---|---|---|
| **4** | Classify each component into one of the following categories:<br>• **Node:** VAX or Alpha system in the OpenVMS Cluster configuration.<br>• **Adapter:** LAN adapter on the system that is normally used for OpenVMS Cluster communications.<br>• **Component:** Generic component in the network. Components in this category can usually be shown to be working if at least one path through them is working. Wiring concentrators, DELNI interconnects, DEMPR repeaters, LAN bridges, and LAN segments and rings typically fall into this category.<br>• **Cloud:** Generic component in the network. Components in this category cannot be shown to be working even if one or more paths are shown to be working. | The cloud component is necessary only when multiple paths exist between two points within the network, such as with redundant bridging between LAN segments. At a high level, multiple paths can exist; however, during operation, this bridge configuration allows only one path to exist at one time. In general, this bridge example is probably better handled by representing the active bridge in the description as a component and ignoring the standby bridge. (You can identify the active bridge with such network monitoring software as RBMS or DECelms.) With the default bridge parameters, failure of the active bridge will be called out. |
| **5** | Use the component labels from step 3 to describe each of the connections in the OpenVMS Cluster communications network. | |
| **6** | Choose a node or group of nodes to run the network failure analysis program. | You should run the program only on a node that you included in the physical description when you edited LAVC$FAILURE_ANALYSIS.MAR. The network failure analysis program on one node operates independently from other systems in the OpenVMS Cluster. So, for executing the network failure analysis program, you should choose systems that are not normally shut down. Other good candidates for running the program are systems with the following characteristics:<br>• Faster CPU speed<br>• Larger amounts of memory<br>• More LAN adapters (running the NISCA protocol)<br>**Note:** The physical description is loaded into nonpaged pool, and all processing is performed at IPL 8. CPU use increases as the average number of network components in the network path increases. CPU use also increases as the total number of network paths increases. |

## D.5.2 Edit the Source File

Follow these steps to edit the LAVC$FAILURE_ANALYSIS.MAR program.

| Step | Action |
|---|---|
| **1** | Copy the following files from SYS$EXAMPLES to your local directory:<br>• LAVC$FAILURE_ANALYSIS.MAR<br>• LAVC$BUILD.COM |
| **2** | Use the OpenVMS Cluster network map and the other information you collected to edit the copy of LAVC$FAILURE_ANALYSIS.MAR. |

Example D–1 shows the portion of LAVC$FAILURE_ANALYSIS.MAR that you edit.

**Example D–1 Portion of LAVC$FAILURE_ANALYSIS.MAR to Edit**

```
;       *** Start edits here ***

;       Edit 1.
;
;               Define the hardware components needed to describe
;               the physical configuration.
;

        NEW_COMPONENT   SYSTEM          NODE
        NEW_COMPONENT   LAN_ADP         ADAPTER
        NEW_COMPONENT   DEMPR           COMPONENT
        NEW_COMPONENT   DELNI           COMPONENT
        NEW_COMPONENT   SEGMENT         COMPONENT
        NEW_COMPONENT   NET_CLOUD       CLOUD


;       Edit 2.
;
;                       Diagram of a multi-adapter local area OpenVMS Cluster
;
;
;        Sa   -------+---------------+---------------+---------------+-------
;                    |               |               |               |
;                    |             MPR_A             |               |
;                    |           .---+---.           |               |
;                    |         1|       1|         1|               |
;                   BrA       ALPHA     BETA       DELTA            BrB
;                    |         2|       2|         2|               |
;                    |          '---+---'           |               |
;                    |            LNI_A              |               |
;                    |              |                |               |
;        Sb   -------+---------------+---------------+---------------+-------
;
;
;       Edit 3.
;
; Label    Node                      Description
; -----   ------  ---------------------------------------------

  SYSTEM  A,      ALPHA,  < - MicroVAX II; In the Computer room> . . .
  LAN_ADP A1,     ,       <XQA; ALPHA - MicroVAX II; Computer room>, . . .
  LAN_ADP A2,     ,       <XQB; ALPHA - MicroVAX II; Computer room>, . . .

  SYSTEM  B,      BETA,   < - MicroVAX 3500; In the Computer room> . . .
  LAN_ADP B1,     ,       <XQA; BETA - MicroVAX 3500; Computer room>, . . .
  LAN_ADP B2,     ,       <XQB; BETA - MicroVAX 3500; Computer room>, . . .
```

(continued on next page)

**Example D–1 (Cont.) Portion of LAVC$FAILURE_ANALYSIS.MAR to Edit**

```
  SYSTEM  D,      DELTA, < - VAXstation II; In Dan's office> . . .
  LAN_ADP D1,     ,       <XQA; DELTA - VAXstation II; Dan's office>, . . .
  LAN_ADP D2,     ,       <XQB; DELTA - VAXstation II; Dan's office>, . . .

;       Edit 4.
;
;               Label each of the other network components.
;

        DEMPR   MPR_A, , <Connected to segment A; In the Computer room>
        DELNI   LNI_A, , <Connected to segment B; In the Computer room>

        SEGMENT Sa,  , <Ethernet segment A>
        SEGMENT Sb,  , <Ethernet segment B>

        NET_CLOUD       BRIDGES, , <Bridging between ethernet segments A and B>

;       Edit 5.
;
;               Describe the network connections.
;

        CONNECTION      Sa,     MPR_A
        CONNECTION              MPR_A,  A1
        CONNECTION                      A1,     A
        CONNECTION              MPR_A,  B1
        CONNECTION                      B1,     B

        CONNECTION      Sa,     D1
        CONNECTION              D1,     D

        CONNECTION      Sa,     BRIDGES
        CONNECTION      Sb,     BRIDGES

        CONNECTION      Sb,     LNI_A
        CONNECTION              LNI_A,  A2
        CONNECTION                      A2,     A
        CONNECTION              LNI_A,  B2
        CONNECTION                      B2,     B

        CONNECTION      Sb,     D2
        CONNECTION              D2,     D

        .PAGE

;       *** End of edits ***
```

In the program, Edit *number* identifies a place where you edit the program to incorporate information about your network. Make the following edits to the program:

| Location | Action |
|---|---|
| Edit 1 | Define a category for each component in the configuration. Use the information from step 5 in Section D.5.1. Use the following format:<br><br>`NEW_COMPONENT  component_type  category`<br><br>**Example:** The following example shows how to define a DEMPR repeater as part of the component category:<br><br>`NEW_COMPONENT  DEMPR  COMPONENT` |
| Edit 2 | Incorporate the network map you drew for step 1 of Section D.5.1. Including the map here in LAVC$FAILURE_ANALYSIS.MAR gives you an electronic record of the map that you can locate and update more easily than a drawing on paper. |

| Location | Action |
|---|---|
| Edit 3 | List each OpenVMS Cluster node and its LAN adapters. Use one line for each node. Each line should include the following information. Separate the items of information with commas to create a table of the information.<br>• Component type, followed by a comma.<br>• Label from the network map, followed by a comma.<br>• Node name (for SYSTEM components only). If there is no node name, enter a comma.<br>• Descriptive text that the network failure analysis program displays if it detects a failure with this component. Put this text within angle brackets (< >). This text should include the component's physical location.<br>• LAN hardware address (for LAN adapters).<br>• DECnet LAN address for the LAN adapter that DECnet uses. |
| Edit 4 | List each of the other network components. Use one line for each component. Each line should include the following information:<br>• Component name and category you defined with NEW_COMPONENT.<br>• Label from the network map.<br>• Descriptive text that the network failure analysis program displays if it detects a failure with this component. Include a description of the physical location of the component.<br>• LAN hardware address (optional).<br>• Alternate LAN address (optional). |
| Edit 5 | Define the connections between the network components. Use the CONNECTION macro and the labels for the two components that are connected. Include the following information:<br>• CONNECTION macro name<br>• First component label<br>• Second component label |

**Reference:** You can find more detailed information about this exercise within the source module SYS$EXAMPLES:LAVC$FAILURE_ANALYSIS.MAR.

### D.5.3 Assemble and Link the Program

Use the following command procedure to assemble and link the program:

```
$ @LAVC$BUILD.COM LAVC$FAILURE_ANALYSIS.MAR
```

Make the edits necessary to fix the assembly or link errors, such as errors caused by mistyping component labels in the path description. Assemble the program again.

### D.5.4 Modify Startup Files

Before you execute the LAVC$FAILURE_ANALYSIS.EXE procedure, modify the startup files to run the procedure only on the node for which you supplied data.

**Example:** To execute the program on node OMEGA, you would modify the startup files in SYS$COMMON:[SYSMGR] to include the following conditional statement:

```
$ If F$GETSYI ("nodename").EQS."OMEGA"
$ THEN
$   RUN SYS$MANAGER:LAVC$FAILURE_ANALYSIS.EXE
$ ENDIF
```

### D.5.5 Execute the Program

To run the LAVC$FAILURE_ANALYSIS.EXE program, follow these steps:

| Step | Action |
|---|---|
| **1** | Use an account that has the PHY_IO privilege. |
| **2** | Execute the program on each of the nodes that will perform the network failure analysis: |
| | `$ RUN SYS$MANAGER:LAVC$FAILURE_ANALYSIS.EXE` |

After it executes, the program displays the approximate amount of nonpaged pool required for the network description. The display is similar to the following:

```
Non-paged Pool Usage:  ~ 10004 bytes
```

### D.5.6 Modify MODPARAMS.DAT

On each system running the network failure analysis, modify the file SYS$SPECIFIC:[SYSEXE]MODPARAMS.DAT to include the following lines, replacing *value* with the value that was displayed for nonpaged pool usage:

```
ADD_NPAGEDYN = value
ADD_NPAGEVIR = value
```

Run AUTOGEN on each system for which you modified MODPARAMS.DAT.

### D.5.7 Test the Program

Test the program by causing a failure. For example, disconnect a transceiver cable or ThinWire segment, or cause a power failure on a bridge, a DELNI interconnect, or a DEMPR repeater. Then check the OPCOM messages to see whether LAVC$FAILURE_ANALYSIS reports the failed component correctly. If it does not report the failure, check your edits to the network failure analysis program.

### D.5.8 Display Suspect Components

When an OpenVMS Cluster network component failure occurs, OPCOM displays a list of suspected components. Displaying the list through OPCOM allows the system manager to enable and disable selectively the display of these messages.

The following are sample displays:

```
%%%%%%%%%%  OPCOM   1-JAN-1994 14:16:13.30  %%%%%%%%%%
(from node BETA at 1-JAN-1994 14:15:55.38)
Message from user SYSTEM on BETA LAVC-W-PSUSPECT, component_name

%%%%%%%%%%  OPCOM   1-JAN-1994 14:16:13.41  %%%%%%%%%%
(from node BETA at 1-JAN-1994 14:15:55.49)
Message from user SYSTEM on BETA %LAVC-W-PSUSPECT, component_name

%%%%%%%%%%  OPCOM   1-JAN-1994 14:16:13.50  %%%%%%%%%%
(from node BETA at 1-JAN-1994 14:15:55.58)
Message from user SYSTEM on BETA %LAVC-I-ASUSPECT, component_name
```

The OPCOM display of suspected failures uses the following prefixes to list suspected failures:

- %LAVC-W-PSUSPECT—Primary suspects
- %LAVC-I-ASUSPECT—Secondary or additional suspects
- %LAVC-S-WORKING—Suspect component is now working

The text following the message prefix is the description of the network component you supplied when you edited LAVC$FAILURE_ANALYSIS.MAR.

# E

# Subroutines for LAN Control

## E.1 Introduction

In addition to the sample programs described in Appendix D, a number of subroutines are provided as a way of extending the capabilities of the sample programs. Table E–1 describes the subroutines.

**Table E–1 Subroutines for LAN Control**

| Subroutine | Description |
|---|---|
| **To manage LAN adapters:** | |
| SYS$LAVC_START_BUS | Directs PEDRIVER to start the NISCA protocol on a specific LAN adapter. |
| SYS$LAVC_STOP_BUS | Directs PEDRIVER to stop the NISCA protocol on a specific LAN adapter. |
| **To control the network failure analysis system:** | |
| SYS$LAVC_DEFINE_NET_COMPONENT | Creates a representation of a physical network component. |
| SYS$LAVC_DEFINE_NET_PATH | Creates a directed list of network components between two network nodes. |
| SYS$LAVC_ENABLE_ANALYSIS | Enables the network failure analysis, which makes it possible to analyze future channel failures. |
| SYS$LAVC_DISABLE_ANALYSIS | Stops the network failure analysis and deallocates the memory used for the physical network description. |

### E.1.1 Purpose of the Subroutines

The subroutines described in this appendix are used by the the LAN control programs, LAVC$FAILURE_ANALYSIS.MAR, LAVC$START_BUS.MAR, and LAVC$STOP_BUS.MAR. Although these programs are sufficient for controlling LAN networks, you may also find it helpful to use the LAN control subroutines to further manage LAN adapters.

## E.2 Starting the NISCA Protocol

The SYS$LAVC_START_BUS subroutine starts the NISCA protocol on a specified LAN adapter. To use the routine SYS$LAVC_START_BUS, specify the following parameter:

| Parameter | Description |
|---|---|
| BUS_NAME | String descriptor representing the LAN adapter name buffer, passed by reference. The LAN adapter name must consist of 15 characters or fewer. |

**Example:** The following Fortran sample program uses SYS$LAVC_START_BUS to start the NISCA protocol on the LAN adapter XQA:

```
PROGRAM START_BUS

EXTERNAL SYS$LAVC_START_BUS
INTEGER*4 SYS$LAVC_START_BUS
INTEGER*4 STATUS

STATUS = SYS$LAVC_START_BUS ( 'XQA0:' )

CALL SYS$EXIT ( %VAL ( STATUS ))

END
```

### E.2.1 Status

The SYS$LAVC_START_BUS subroutine returns a status value in register R0, as described in Table E–2.

**Table E–2 SYS$LAVC_START_BUS Status**

| Status | Result |
|---|---|
| Success | Indicates that PEDRIVER is attempting to start the NISCA protocol on the specified adapter. |
| Failure | Indicates that PEDRIVER cannot start the protocol on the specified LAN adapter. |

### E.2.2 Error Messages

SYS$LAVC_START_BUS can return the error condition codes shown in the following table.

| Condition Code | Description |
|---|---|
| SS$_ACCVIO | This status is returned for the following conditions:<br>• No access to the argument list<br>• No access to the LAN adapter name buffer descriptor<br>• No access to the LAN adapter name buffer |
| SS$_DEVACTIVE | Bus already exists. PEDRIVER is already trying to use this LAN adapter for the NISCA protocol. |
| SS$_INSFARG | Not enough arguments supplied. |
| SS$_INSFMEM | Insufficient nonpaged pool to create the bus data structure. |
| SS$_INVBUSNAM | Invalid bus name specified. The device specified does not represent a LAN adapter that can be used for the protocol. |
| SS$_IVBUFLEN | This status value is returned under the following conditions:<br>• The LAN adapter name contains no characters (length = 0).<br>• The LAN adapter name contains more than 15 characters. |

| Condition Code | Description |
|---|---|
| SS$_NOSUCHDEV | This status value is returned under the following conditions:<br>• The LAN adapter name specified does not correspond to a LAN device available to PEDRIVER on this system.<br>• No LAN drivers are loaded in this system; the value for NET$AR_LAN_VECTOR is 0.<br>• PEDRIVER is not initialized; PEDRIVER's PORT structure is not available.<br>**Note:** By calling this routine, an error-log message may be generated. |
| SS$_NOTNETDEV | PEDRIVER does not support the specified LAN device. |
| SS$_SYSVERDIF | The specified LAN device's driver does not support the VCI interface version required by PEDRIVER. |

PEDRIVER can return additional errors that indicate it has failed to create the connection to the specified LAN adapter.

## E.3 Stopping the NISCA Protocol

The SYS$LAVC_STOP_BUS routine stops the NISCA protocol on a specific LAN adapter.

**Caution:** Stopping the NISCA protocol on all LAN adapters causes satellites to hang and could cause cluster systems to fail with a CLUEXIT bugcheck.

To use this routine, specify the parameter described in the following table.

| Parameter | Description |
|---|---|
| BUS_NAME | String descriptor representing the LAN adapter name buffer, passed by reference. The LAN adapter name must consist of 15 characters or fewer. |

**Example:** The following Fortran sample program shows how SYS$LAVC_STOP_BUS is used to stop the NISCA protocol on the LAN adapter XQB:

```
PROGRAM STOP_BUS

EXTERNAL SYS$LAVC_STOP_BUS
INTEGER*4 SYS$LAVC_STOP_BUS
INTEGER*4 STATUS

STATUS = SYS$LAVC_STOP_BUS ( 'XQB' )

CALL SYS$EXIT ( %VAL ( STATUS ))

END
```

### E.3.1 Status

The SYS$LAVC_STOP_BUS subroutine returns a status value in register R0, as described in Table E–3.

**Table E–3 SYS$LAVC_STOP_BUS Status**

| Status | Result |
|---|---|
| Success | Indicates that PEDRIVER is attempting to shut down the NISCA protocol on the specified adapter. |
| Failure | Indicates that PEDRIVER cannot shut down the protocol on the specified LAN adapter. However, PEDRIVER performs the shutdown asynchronously, and there could be other reasons why PEDRIVER is unable to complete the shutdown. |

When the LAVC$STOP_BUS module executes successfully, the following device-attention entry is written to the system error log:

```
DEVICE ATTENTION . . .
NI-SCS SUB-SYSTEM . . .
FATAL ERROR DETECTED BY DATALINK . . .
```

In addition, the following hexadecimal values are written to the STATUS field of the entry:

First longword (00000001)
Second longword (00001201)

This error-log entry indicates expected behavior and can be ignored. However, if the first longword of the STATUS field contains a value other than hexadecimal value 00000001, an error has occurred and further investigation may be necessary.

## E.3.2 Error Messages

SYS$LAVC_STOP_BUS can return the error condition codes shown in the following table.

| Condition Code | Description |
|---|---|
| SS$_ACCVIO | This status is returned for the following conditions:<br>• No access to the argument list<br>• No access to the LAN adapter name buffer descriptor<br>• No access to the LAN adapter name buffer |
| SS$_INVBUSNAM | Invalid bus name specified. The device specified does not represent a LAN adapter that can be used for the NISCA protocol. |
| SS$_IVBUFLEN | This status value is returned under the following conditions:<br>• The LAN adapter name contains no characters (length = 0).<br>• The LAN adapter name has more than 15 characters. |

| Condition Code | Description |
|---|---|
| SS$_NOSUCHDEV | This status value is returned under the following conditions:<br>• The LAN adapter name specified does not correspond to a LAN device that is available to PEDRIVER on this system.<br>• No LAN drivers are loaded in this system. NET$AR_LAN_VECTOR is zero.<br>• PEDRIVER is not initialized. PEDRIVER's PORT structure is not available. |

## E.4 Creating a Representation of a Network Component

The SYS$LAVC_DEFINE_NET_COMPONENT subroutine creates a representation for a physical network component.

Use the following format to specify the parameters:

```
STATUS = SYS$LAVC_DEFINE_NET_COMPONENT (
                        component_description,
                        nodename_length,
                        component_type,
                        lan_hardware_addr,
                        lan_decnet_addr,
                        component_id_value )
```

Table E–4 describes the SYS$LAVC_DEFINE_NET_COMPONENT parameters.

**Table E–4 SYS$LAVC_DEFINE_NET_COMPONENT Parameters**

| Parameter | Description |
|---|---|
| component_description | Address of a string descriptor representing network component name buffer. The length of the network component name must be less than or equal to the number of COMP$C_MAX_NAME_LEN characters. |
| nodename_length | Address of the length of the node name. This address is located at the beginning of the network component name buffer for COMP$C_NODE types. You should use zero for other component types. |
| component_type | Address of the component type. These values are defined by $PEMCOMPDEF, found in SYS$LIBRARY:LIB.MLB. |
| lan_hardware_addr | Address of a string descriptor of a buffer containing the component's LAN hardware address (6 bytes). You must specify this value for COMP$C_ADAPTER types. For other component types, this value is optional. |
| lan_decnet_addr | String descriptor of a buffer containing the component's LAN DECnet address (6 bytes). This is an optional parameter for all component types. |
| component_id_value | Address of a longword that is written with the component ID value. |

### E.4.1 Status

If successful, the SYS$LAVC_DEFINE_NET_COMPONENT subroutine creates a COMP data structure and returns its ID value. This subroutine copies user-specified parameters into the data structure and sets the reference count to zero.

The component ID value is a 32-bit value that has a one-to-one association with a network component. Lists of these component IDs are passed to SYS$LAVC_DEFINE_NET_PATH to specify the components used when a packet travels from one node to another.

### E.4.2 Error Messages

SYS$LAVC_DEFINE_NET_COMPONENT can return the error condition codes shown in the following table.

| Condition Code | Description |
|---|---|
| SS$_ACCVIO | This status is returned for the following conditions:<br>• No access to the network component name buffer descriptor<br>• No access to the network component name buffer<br>• No access to the component's LAN hardware address if a nonzero value was specified<br>• No access to the component's LAN DECnet address if a nonzero value was specified<br>• No access to the lan_hardware_addr string descriptor<br>• No access to the lan_decnet_addr string descriptor<br>• No write access to the component_id_value address<br>• No access to the component_type address<br>• No access to the nodename_length address<br>• No access to the argument list |
| SS$_DEVACTIVE | Analysis program already running. You must stop the analysis by calling the SYS$LAVC_DISABLE_ANALYSIS before you define the network components and the network component lists. |
| SS$_INSFARG | Not enough arguments supplied. |
| SS$_INVCOMPTYPE | The component type is either 0 or greater than or equal to COMP$C_INVALID. |

| Condition Code | Description |
|---|---|
| SS$_IVBUFLEN | This status value is returned under the following conditions:<br>• The component name has no characters (length = 0).<br>• Length of the component name is greater than COMP$C_MAX_NAME_LEN.<br>• The node name has no characters (length = 0) and the component type is COMP$C_NODE.<br>• The node name has more than 8 characters and the component type is COMP$C_NODE.<br>• The lan_hardware_addr string descriptor has fewer than 6 characters.<br>• The lan_decnet_addr has fewer than 6 characters. |

## E.5 Creating a Network Component List

The SYS$LAVC_DEFINE_NET_PATH subroutine creates a directed list of network components between two network nodes. A **directed list** is a list of all the components through which a packet passes as it travels from the failure analysis node to other nodes in the cluster network.

Use the following format to specify the parameters:

```
STATUS = SYS$LAVC_DEFINE_NET_PATH (
                        network_component_list,
                        used_for_analysis_status,
                        bad_component_id )
```

Table E–5 describes the SYS$LAVC_DEFINE_NET_PATH parameters.

**Table E–5 SYS$LAVC_DEFINE_NET_PATH Parameters**

| Parameter | Description |
|---|---|
| network_component_list | Address of a string descriptor for a buffer containing the component ID values for each of the components in the path. List the component ID values in the order in which a network message travels through them. Specify components in the following order:<br>1. Local node<br>2. Local LAN adapter<br>3. Intermediate network components<br>4. Remote network LAN adapter<br>5. Remote node<br>You must list two nodes and two LAN adapters in the network path. The buffer length must be greater than 15 bytes and less than 509 bytes. |

(continued on next page)

**Table E–5 (Cont.) SYS$LAVC_DEFINE_NET_PATH Parameters**

| Parameter | Description |
|---|---|
| used_for_analysis_status | Address of a longword status value that is written. This status indicates whether this network path has any value for the network failure analysis. |
| bad_component_id | Address of a longword value that contains the erroneous component ID if an error is detected while processing the component list. |

## E.5.1 Status

This subroutine creates a directed list of network components that describe a specific network path. If SYS$LAVC_DEFINE_NET_PATH is successful, it creates a CLST data structure. If one node is the local node, then this data structure is associated with a PEDRIVER channel. In addition, the reference count for each network component in the list is incremented. If neither node is the local node, then the used_for_analysis_status address contains an error status.

The SYS$LAVC_DEFINE_NET_PATH subroutine returns a status value in register R0, as described in Table E–6, indicating whether the network component list has the correct construction.

**Table E–6 SYS$LAVC_DEFINE_NET_PATH Status**

| Status | Result |
|---|---|
| Success | The used_for_analysis_status value indicates whether the network path is useful for network analysis performed on the local node. |
| Failure | If a failure status returned in R0 is SS$_INVCOMPID, the bad_component_id address contains the value of the bad_component_id found in the buffer. |

## E.5.2 Error Messages

SYS$LAVC_DEFINE_NET_PATH can return the error condition codes shown in the following table.

| Condition Code | Description |
|---|---|
| SS$_ACCVIO | This status value can be returned under the following conditions:<br>• No access to the descriptor or the network component ID value buffer<br>• No access to the argument list<br>• No write access to the used_for_analysis_status address<br>• No write access to the bad_component_id address |
| SS$_DEVACTIVE | Analysis already running. You must stop the analysis by calling the SYS$LAVC_DISABLE_ANALYSIS function before defining the network components and the network component lists. |
| SS$_INSFARG | Not enough arguments supplied. |
| SS$_INVCOMPID | Invalid network component ID specified in the buffer. The bad_component_id address contains the failed component ID. |

| Condition Code | Description |
|---|---|
| SS$_INVCOMPLIST | This status value can be returned under the following conditions:<br>• Fewer than two nodes were specified in the node list.<br>• More than two nodes were specified in the list.<br>• The first network component ID was not a COMP$C_NODE type.<br>• The last network component ID was not a COMP$C_NODE type.<br>• Fewer than two adapters were specified in the list.<br>• More than two adapters were specified in the list. |
| SS$_IVBUFLEN | Length of the network component ID buffer is less than 16, is not a multiple of 4, or is greater than 508. |
| SS$_RMTPATH | Network path is not associated with the local node. This status is returned only to indicate whether this path was needed for network failure analysis on the local node. |

## E.6 Starting Network Component Failure Analysis

The SYS$LAVC_ENABLE_ANALYSIS subroutine starts the network component failure analysis.

**Example:** The following is an example of using the SYS$LAVC_ENABLE_ANALYSIS subroutine:

```
STATUS = SYS$LAVC_ENABLE_ANALYSIS ( )
```

### E.6.1 Status

This subroutine attempts to enable the network component failure analysis code. The attempt will succeed if at least one component list is defined.

SYS$LAVC_ENABLE_ANALYSIS returns a status in register R0.

### E.6.2 Error Messages

SYS$LAVC_ENABLE_ANALYSIS can return the error condition codes shown in the following table.

| Condition Code | Description |
|---|---|
| SS$_DEVOFFLINE | PEDRIVER is not properly initialized. ROOT or PORT block is not available. |
| SS$_NOCOMPLSTS | No network connection lists exist. Network analysis is not possible. |
| SS$_WASSET | Network component analysis is already running. |

# E.7 Stopping Network Component Failure Analysis

The SYS$LAVC_DISABLE_ANALYSIS subroutine stops the network component failure analysis.

**Example:** The following is an example of using SYS$LAVC_DISABLE_ANALYSIS:

```
STATUS = SYS$LAVC_DISABLE_ANALYSIS ( )
```

This subroutine disables the network component failure analysis code and, if analysis was enabled, deletes all the network component definitions and network component list data structures from nonpaged pool.

## E.7.1 Status

SYS$LAVC_DISABLE_ANALYSIS returns a status in register R0.

## E.7.2 Error Messages

SYS$LAVC_DISABLE_ANALYSIS can return the error condition codes shown in the following table.

| Condition Code | Description |
|---|---|
| SS$_DEVOFFLINE | PEDRIVER is not properly initialized. ROOT or PORT block is not available. |
| SS$_WASCLR | Network component analysis already stopped. |

# F

# Troubleshooting the NISCA Protocol

NISCA is the transport protocol responsible for carrying messages, such as disk I/Os and lock messages, across Ethernet and FDDI LANs to other nodes in the cluster. The acronym NISCA refers to the protocol that implements an Ethernet or FDDI network interconnect (NI) according to the System Communications Architecture (SCA).

Using the NISCA protocol, an OpenVMS software interface emulates the CI port interface—that is, the software interface is identical to that of the CI bus, except that data is transferred over a LAN. The NISCA protocol allows OpenVMS Cluster communication over the LAN without the need for any special hardware.

This appendix describes the NISCA transport protocol and provides troubleshooting strategies to help a network manager pinpoint network-related problems. Because troubleshooting hard component failures in the LAN is best accomplished using a LAN analyzer, this appendix also describes the features and setup of a LAN analysis tool.

**Note**

Additional troubleshooting information specific to the revised PEDRIVER is planned for the next revision of this manual.

## F.1 How NISCA Fits into the SCA

The NISCA protocol is an implementation of the Port-to-Port Driver (PPD) protocol of the SCA.

### F.1.1 SCA Protocols

As described in Chapter 2, the SCA is a software architecture that provides efficient communication services to low-level distributed applications (for example, device drivers, file services, network managers).

The SCA specifies a number of protocols for OpenVMS Cluster systems, including System Applications (SYSAP), System Communications Services (SCS), the Port-to-Port Driver (PPD), and the Physical Interconnect (PI) of the device driver and LAN adapter. Figure F–1 shows these protocols as interdependent levels that make up the SCA architecture. Figure F–1 shows the NISCA protocol as a particular implementation of the PPD layer of the SCA architecture.

**Figure F–1 Protocols in the SCA Architecture**

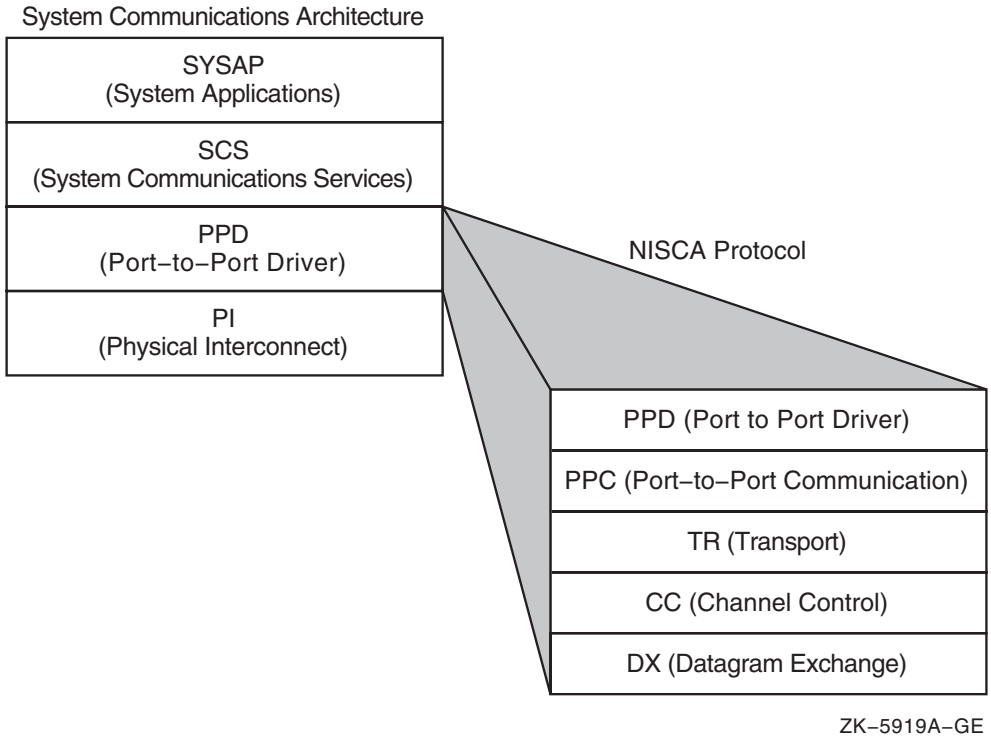



Table F–1 describes the levels of the SCA protocol shown in Figure F–1.

**Table F–1 SCA Protocol Layers**

| Protocol | Description |
|---|---|
| SYSAP | Represents clusterwide system applications that execute on each node. These system applications share communication paths in order to send messages between nodes. Examples of system applications are disk class drivers (such as DUDRIVER), the MSCP server, and the connection manager. |
| SCS | Manages connections around the OpenVMS Cluster and multiplexes messages between system applications over a common transport called a **virtual circuit** (see Section F.1.2). The SCS layer also notifies individual system applications when a connection fails so that they can respond appropriately. For example, an SCS notification might trigger DUDRIVER to fail over a disk, trigger a cluster state transition, or notify the connection manager to start timing reconnect (RECNXINTERVAL) intervals. |

(continued on next page)

**Table F–1 (Cont.) SCA Protocol Layers**

| Protocol | Description |
|---|---|
| PPD | Provides a message delivery service to other nodes in the OpenVMS Cluster system. |

| PPD Level | Description |
|---|---|
| Port-to-Port Driver (PPD) | Establishes virtual circuits and handles errors. |
| Port-to-Port Communication (PPC) | Provides port-to-port communication, datagrams, sequenced messages, and block transfers. "Segmentation" also occurs at the PPC level. Segmentation of large blocks of data is done differently on a LAN than on a CI or a DSSI bus. LAN data packets are fragmented according to the size allowed by the particular LAN communications path, as follows: |

| Port-to-Port Communications | Packet Size Allowed |
|---|---|
| Ethernet-to-Ethernet | 1498 bytes |
| FDDI-to-Ethernet | 1498 bytes |
| FDDI-to-Ethernet-to-FDDI | 1498 bytes |
| FDDI-to-FDDI | 4468 bytes |
| **Note:** The default value is 1498 bytes for both Ethernet and FDDI. | |

| | |
|---|---|
| Transport (TR) | Provides an error-free path, called a virtual circuit (see Section F.1.2), between nodes. The PPC level uses a virtual circuit for transporting sequenced messages and datagrams between two nodes in the cluster. |
| Channel Control (CC) | Manages network paths, called channels, between nodes in an OpenVMS Cluster. The CC level maintains channels by sending HELLO datagram messages between nodes. A node sends a HELLO datagram messages to indicate it is still functioning. The TR level uses channels to carry virtual circuit traffic. |
| Datagram Exchange (DX) | Interfaces to the LAN driver. |

PI — Provides connections to LAN devices. PI represents LAN drivers and adapters over which packets are sent and received.

| PI Component | Description |
|---|---|
| LAN drivers | Multiplex NISCA and many other clients (such as DECnet, TCP/IP, LAT, LAD/LAST) and provide them with datagram services on Ethernet and FDDI network interfaces. |
| LAN adapters | Consist of the LAN network driver and adapter hardware. |

### F.1.2 Paths Used for Communication

The NISCA protocol controls communications over the paths described in Table F–2.

**Table F–2 Communication Paths**

| Path | Description |
|---|---|
| Virtual circuit | A common transport that provides reliable port-to-port communication between OpenVMS Cluster nodes in order to:<br>• Ensure the delivery of messages without duplication or loss, each port maintains a virtual circuit with every other remote port.<br>• Ensure the sequential ordering of messages, virtual circuit sequence numbers are used on the individual packets. Each transmit message carries a sequence number; duplicates are discarded.<br>The virtual circuit descriptor table in each port indicates the status of it's port-to-port circuits. After a virtual circuit is formed between two ports, communication can be established between SYSAPs in the nodes. |
| Channel | A logical communication path between two LAN adapters located on different nodes. Channels between nodes are determined by the pairs of adapters and the connecting network. For example, two nodes, each having two adapters, could establish four channels. The messages carried by a particular virtual circuit can be sent over any of the channels connecting the two nodes. |

**Note:** The difference between a channel and a virtual circuit is that channels provide a path for datagram service. Virtual circuits, layered on channels, provide an error-free path between nodes. Multiple channels can exist between nodes in an OpenVMS Cluster but only one virtual circuit can exist between any two nodes at a time.

### F.1.3 PEDRIVER

The port emulator driver, PEDRIVER, implements the NISCA protocol and establishes and controls channels for communication between local and remote LAN ports.

PEDRIVER implements a packet delivery service (at the TR level of the NISCA protocol) that guarantees the sequential delivery of messages. The messages carried by a particular virtual circuit can be sent over any of the channels connecting two nodes. The choice of channel is determined by the sender (PEDRIVER) of the message. Because a node sending a message can choose any channel, PEDRIVER, as a receiver, must be prepared to receive messages over any channel.

At any point in time, the TR level makes use of a single “preferred channel” to carry the traffic for a particular virtual circuit.

**Reference:** See Appendix G for more information about how transmit channels are selected.

## F.2 Addressing LAN Communication Problems

This section describes LAN Communication Problems and how to address them.

### F.2.1 Symptoms

Communication trouble in OpenVMS Cluster systems may be indicated by symptoms such as the following:

- Poor performance
- Console messages
  - "Virtual circuit closed" messages from PEA0 (PEDRIVER) on the console
  - "Connection loss" OPCOM messages on the console
  - CLUEXIT bugchecks
  - "Excessive packet losses on LAN Path" messages on the console
- Repeated loss of a virtual circuit or multiple virtual circuits over a short period of time (fewer than 10 minutes)

Before you initiate complex diagnostic procedures, do not overlook the obvious. Always make sure the hardware is configured and connected properly and that the network is started. Also, make sure system parameters are set correctly on all nodes in the OpenVMS Cluster.

### F.2.2 Traffic Control

Keep in mind that an OpenVMS Cluster system generates substantially heavier traffic than other LAN protocols. In many cases, cluster behavior problems that appear to be related to the network might actually be related to software, hardware, or user errors. For example, a large amount of traffic does not necessarily indicate a problem with the OpenVMS Cluster network. The amount of traffic generated depends on how the users utilize the system and the way that the OpenVMS Cluster is configured with additional interconnects (such as DSSI and CI).

If the amount of traffic generated by the OpenVMS Cluster exceeds the expected or desired levels, then you might be able to reduce the level of traffic by:

- Adding DSSI or CI interconnects
- Shifting the user load between machines
- Adding LAN segments and reconfiguring the LAN connections across the OpenVMS Cluster system

### F.2.3 Excessive Packet Losses on LAN Paths

Prior to OpenVMS Version 7.3, an SCS virtual circuit closure was the first indication that a LAN path had become unusable. In OpenVMS Version 7.3, whenever the last usable LAN path is losing packets at an excessive rate, PEDRIVER displays the following console message:

```
%PEA0, Excessive packet losses on LAN path from local-device-name
 to device-name on REMOTE NODE node-name
```

This message is displayed when PEDRIVER recently had to perform an excessively high rate of packet retransmissions on the LAN path consisting of the local device, the intervening network, and the device on the remote node. The message indicates that the LAN path has degraded and is approaching, or has reached, the point where reliable communications with the remote node are no longer possible. It is likely that the virtual circuit to the remote node will close if the losses continue. Furthermore, continued operation with high LAN packet losses can result in significant loss in performance because of the communication delays resulting from the packet loss detection timeouts and packet retransmission.

The corrective steps to take are:

1. Check the local and remote LAN device error counts to see whether a problem exists on the devices. Issue the following commands on each node:

   ```
   $ SHOW DEVICE local-device-name
   $ MC SCACP
   SCACP> SHOW LAN device-name
   $ MC LANCP
   LANCP> SHOW DEVICE device-name/COUNT
   ```

2. If device error counts on the local devices are within normal bounds, contact your network administrators to request that they diagnose the LAN path between the devices.

### F.2.4 Preliminary Network Diagnosis

If the symptoms and preliminary diagnosis indicate that you might have a network problem, troubleshooting LAN communication failures should start with the step-by-step procedures described in Appendix C. Appendix C helps you diagnose and solve common Ethernet and FDDI LAN communication failures during the following stages of OpenVMS Cluster activity:

- When a computer or a satellite fails to boot
- When a computer fails to join the OpenVMS Cluster
- During run time when startup procedures fail to complete
- When a OpenVMS Cluster hangs

The procedures in Appendix C require that you verify a number of parameters during the diagnostic process. Because system parameter settings play a key role in effective OpenVMS Cluster communications, Section F.2.6 describes several system parameters that are especially important to the timing of LAN bridges, disk failover, and channel availability.

### F.2.5 Tracing Intermittent Errors

Because PEDRIVER communication is based on channels, LAN network problems typically fall into these areas:

- Channel formation and maintenance

  Channels are formed when HELLO datagram messages are received from a remote system. A failure can occur when the HELLO datagram messages are not received or when the channel control message contains the wrong data.

- Retransmission

  A well-configured OpenVMS Cluster system should not perform excessive retransmissions between nodes. Retransmissions between any nodes that occur more frequently than once every few seconds deserve network investigation.

Diagnosing failures at this level becomes more complex because the errors are usually intermittent. Moreover, even though PEDRIVER is aware when a channel is unavailable and performs error recovery based on this information, it does not provide notification when a channel failure occurs; PEDRIVER provides notification only for virtual circuit failures.

However, the Local Area OpenVMS Cluster Network Failure Analysis Program (LAVC$FAILURE_ANALYSIS), available in SYS$EXAMPLES, can help you use PEDRIVER information about channel status. The LAVC$FAILURE_ANALYSIS program (documented in Appendix D) analyzes long-term channel outages, such as hard failures in LAN network components that occur during run time.

This program uses tables in which you describe your LAN hardware configuration. During a channel failure, PEDRIVER uses the hardware configuration represented in the table to isolate which component might be causing the failure. PEDRIVER reports the suspected component through an OPCOM display. You can then isolate the LAN component for repair or replacement.

**Reference:** Section F.7 addresses the kinds of problems you might find in the NISCA protocol and provides methods for diagnosing and solving them.

### F.2.6 Checking System Parameters

Table F–3 describes several system parameters relevant to the recovery and failover time limits for LANs in an OpenVMS Cluster.

**Table F–3 System Parameters for Timing**

| Parameter | Use |
|---|---|
| **RECNXINTERVAL** | |
| Defines the amount of time to wait before removing a node from the OpenVMS Cluster after detection of a virtual circuit failure, which could result from a LAN bridge failure. | If your network uses multiple paths and you want the OpenVMS Cluster to survive failover between LAN bridges, make sure the value of RECNXINTERVAL is greater than the time it takes to fail over those paths.<br>**Reference:** The formula for calculating this parameter is discussed in Section 3.4.7. |
| **MVTIMEOUT** | |
| Defines the amount of time the OpenVMS operating system tries to recover a path to a disk before returning failure messages to the application. | Relevant when an OpenVMS Cluster configuration is set up to serve disks over either the Ethernet or FDDI. MVTIMEOUT is similar to RECNXINTERVAL except that RECNXINTERVAL is CPU to CPU, and MVTIMEOUT is CPU to disk. |

(continued on next page)

**Table F–3 (Cont.) System Parameters for Timing**

| Parameter | Use |
|---|---|
| **SHADOW_MBR_TIMEOUT** | |
| Defines the amount of time that the Volume Shadowing for OpenVMS tries to recover from a transient disk error on a single member of a multiple-member shadow set. | SHADOW_MBR_TIMEOUT differs from MVTIMEOUT because it removes a failing shadow set member quickly. The remaining shadow set members can recover more rapidly once the failing member is removed. |

**Note:** The TIMVCFAIL system parameter, which optimizes the amount of time needed to detect a communication failure, is not recommended for use with LAN communications. This parameter is intended for CI and DSSI connections. PEDRIVER (which is for Ethernet and FDDI) usually surpasses the detection provided by TIMVCFAIL with the listen timeout of 8 to 9 seconds.

## F.2.7 Channel Timeouts

Channel timeouts are detected by PEDRIVER as described in Table F–4.

**Table F–4 Channel Timeout Detection**

| PEDRIVER Actions | Comments |
|---|---|
| Listens for HELLO datagram messages, which are sent over channels at least once every 3 seconds | Every node in the OpenVMS Cluster multicasts HELLO datagram messages on each LAN adapter to notify other nodes that it is still functioning. Receiving nodes know that the network connection is still good. |
| Closes a channel when HELLO datagrams or sequenced messages have not been received for a period of 8 to 9 seconds | Because HELLO datagram messages are transmitted at least once every 3 seconds, PEDRIVER times out a channel only if at least two HELLO datagram messages are lost and there is no sequenced message traffic. |
| Closes a virtual circuit when:<br>• No channels are available.<br>• The packet size of the only available channels is insufficient. | The virtual circuit is not closed if any other channels to the node are available except when the packet sizes of available channels are smaller than the channel being used for the virtual circuit. For example, if a channel fails over from FDDI to Ethernet, PEDRIVER may close the virtual circuit and then reopen it after negotiating the smaller packet size that is necessary for Ethernet segmentation. |
| Does not report errors when a channel is closed | OPCOM “Connection loss” errors or SYSAP messages are not sent to users or other system applications until after the virtual circuit shuts down. This fact is significant, especially if there are multiple paths to a node and a LAN hardware failure occurs. In this case, you might not receive an error message; PEDRIVER continues to use the virtual circuit over another available channel. |
| Reestablishes a virtual circuit when a channel becomes available again | PEDRIVER reopens a channel when HELLO datagram messages are received again. |

# F.3 Using SDA to Monitor LAN Communications

This section describes how to use SDA to monitor LAN communications.

## F.3.1 Isolating Problem Areas

If your system shows symptoms of intermittent failures during run time, you need to determine whether there is a network problem or whether the symptoms are caused by some other activity in the system.

Generally, you can diagnose problems in the NISCA protocol or the network using the OpenVMS System Dump Analyzer utility (SDA). SDA is an effective tool for isolating problems on specific nodes running in the OpenVMS Cluster system.

**Reference:** The following sections describe the use of some SDA commands and qualifiers. You should also refer to the *OpenVMS Alpha System Analysis Tools Manual* or the *OpenVMS VAX System Dump Analyzer Utility Manual* for complete information about SDA for your system.

## F.3.2 SDA Command SHOW PORT

The SDA command SHOW PORT provides relevant information that is useful in troubleshooting PEDRIVER and LAN adapters in particular. Begin by entering the SHOW PORT command, which causes SDA to define cluster symbols. Example F–1 illustrates how the SHOW PORT command provides a summary of OpenVMS Cluster data structures.

**Example F–1 SDA Command SHOW PORT Display**

```
$ ANALYZE/SYSTEM
SDA> SHOW PORT

VAXcluster data structures
--------------------------

                 --- PDT Summary Page ---

 PDT Address          Type         Device          Driver Name
 -----------          ----         -------         -----------

  80C3DBA0             pa          PAA0            PADRIVER
  80C6F7A0             pe          PEA0            PEDRIVER
```

## F.3.3 Monitoring Virtual Circuits

To examine information about the virtual circuit (VC) that carries messages between the local node (where you are running SDA) and another remote node, enter the SDA command SHOW PORT/VC=VC_*remote-node-name*. Example F–2 shows how to examine information about the virtual channel running between a local node and the remote node, NODE11.

**Example F–2 SDA Command SHOW PORT/VC Display**

```
SDA> SHOW PORT/VC=VC_NODE11

VAXcluster data structures
--------------------------
                --- Virtual Circuit (VC) 98625380 ---
Remote System Name:  NODE11  (0:VAX)     Remote SCSSYSTEMID:  19583
Local System ID:  217 (D9)               Status: 0005 open,path
------ Transmit -------  ----- VC Closures -----  (7)--- Congestion Control ----
Msg Xmt(1)      46193196  SeqMsg TMO            0  Pipe Quota/Slo/Max(8) 31/ 7/31
  Unsequence           3  CC DFQ Empty          0  Pipe Quota Reached(9)   213481
  Sequence      41973703  Topology Change(5)    0  Xmt C/T(10)              0/1984
  ReXmt(2)        128/106  NPAGEDYN Low(6)       0  RndTrp uS(11)        18540+7764
  Lone ACK        4219362                          UnAcked Msgs                 0
Bytes Xmt     137312089                          CMD Queue Len/Max         0/21
------- Receive --------  - Messages  Discarded -  ----- Channel Selection -----
Msg Rcv(3)      47612604  No Xmt Chan           0  Preferred Channel    9867F400
  Unsequence           3  Rcv Short Msg         0  Delay Time           FAAD63E0
  Sequence      37877271  Illegal Seq Msg       0  Buffer Size              1424
  ReRcv(4)         13987  Bad Checksum          0  Channel Count              18
  Lone ACK       9721030  TR DFQ Empty          0  Channel Selections      32138
  Cache              314  TR MFQ Empty          0  Protocol                1.3.0
  Ill ACK              0  CC MFQ Empty          0  Open(12) 8-FEB-1994 17:00:05.12
Bytes Rcv    3821742649  Cache Miss            0  Cls(13) 17-NOV-1858 00:00:00.00
```

The SHOW PORT/VC=VC_*remote-node-name* command displays a number of performance statistics about the virtual circuit for the target node. The display groups the statistics into general categories that summarize such things as packet transmissions to the remote node, packets received from the remote node, and congestion control behavior. The statistics most useful for problem isolation are called out in Example F–2 and described in Table F–5.

**Note:** The counters shown in Example F–2 are stored in fixed-size fields and are automatically reset to 0 when a field reaches its maximum value (or when the system is rebooted). Because fields have different maximum sizes and growth rates, the field counters are likely to reset at different times. Thus, for a system that has been running for a long time, some field values may seem illogical and appear to contradict others.

**Table F–5 SHOW PORT/VC Display**

| Field | Description |
|---|---|
| ❶ Msg Xmt (messages transmitted) | Shows the total number of packets transmitted over the virtual circuit to the remote node, including both sequenced and unsequenced (channel control) messages, and lone acknowledgments. (All application data is carried in sequenced messages.) The counters for sequenced messages and lone acknowledgments grow more quickly than most other fields. |
| ❷ ReXmt (retransmission) | Indicates the number of retransmissions and retransmit related timeouts for the virtual circuit.<br>• The rightmost number (106) in the ReXmt field indicates the number of times a timeout occured. A timeout indicates one of the following problems:<br>– The remote system NODE11 did not receive the sequenced message sent by UPNVMS.<br>– The sequenced message arrived but was delayed in transit to NODE11.<br>– The local system UPNVMS did not receive the acknowledgment to the message sent to remote node NODE11.<br>– The acknowledgment arrived but was delayed in transit from NODE11.<br>Congestion either in the network or at one of the nodes can cause the following problems:<br>– Congestion in the network can result in delayed or lost packets. Network hardware problems can also result in lost packets.<br>– Congestion in UPNVMS or NODE11 can result either in packet delay because of queuing in the adapter or in packet discard because of insufficient buffer space.<br>• The leftmost number (128) indicates the number of packets actually retransmitted. For example, if the network loses two packets at the same time, one timeout is counted but two packets are retransmitted. A retransmission occurs when the local node does not receive an acknowledgment for a transmitted packet within a predetermined timeout interval.<br>Although you should expect to see a certain number of retransmissions (especially in heavily loaded networks), an excessive number of retransmissions wastes network bandwidth and indicates excessive load or intermittent hardware failure. If the leftmost value in the ReXmt field is greater than about 0.01% to 0.05% of the total number of the transmitted messages shown in the Msg Xmt field, the OpenVMS Cluster system probably is experiencing excessive network problems or local loss from congestion. |
| ❸ Msg Rcv (messages received) | Indicates the total number of messages received by local node UPNVMS over this virtual circuit. The values for sequenced messages and lone acknowledgments usually increase at a rapid rate. |

(continued on next page)

**Table F–5 (Cont.) SHOW PORT/VC Display**

| Field | Description |
|---|---|
| ❹ ReRcv (rereceive) | Displays the number of packets received redundantly by this system. A remote system may retransmit packets even though the local node has already successfully received them. This happens when the cumulative delay of the packet and its acknowledgment is longer than the estimated round-trip time being used as a timeout value by the remote node. Therefore, the remote node retransmits the packet even though it is unnnecessary.<br><br>Underestimation of the round-trip delay by the remote node is not directly harmful, but the retransmission and subsequent congestion-control behavior on the remote node have a detrimental effect on data throughput. Large numbers indicate frequent bursts of congestion in the network or adapters leading to excessive delays. If the value in the ReRcv field is greater than approximately 0.01% to 0.05% of the total messages received, there may be a problem with congestion or network delays. |
| ❺ Topology Change | Indicates the number of times PEDRIVER has performed a failover from FDDI to Ethernet, which necessitated closing and reopening the virtual circuit. In Example F–2, there have been no failovers. However, if the field indicates a number of failovers, a problem may exist on the FDDI ring. |
| ❻ NPAGEDYN (nonpaged dynamic pool) | Displays the number of times the virtual circuit was closed because of a pool allocation failure on the local node. If this value is nonzero, you probably need to increase the value of the NPAGEDYN system parameter on the local node. |
| ❼ Congestion Control | Displays information about the virtual circuit to control the pipe quota (the number of messages that can be sent to the remote node [put into the "pipe"] before receving an acknowledgment and the retransmission timeout). PEDRIVER varies the pipe quota and the timeout value to control the amount of network congestion. |
| ❽ Pipe Quota/Slo/Max | Indicates the current thresholds governing the pipe quota.<br><br>• The leftmost number (31) is the current value of the pipe quota (transmit window). After a timeout, the pipe quota is reset to 1 to decrease congestion and is allowed to increase quickly as acknowledgments are received.<br><br>• The middle number (7) is the slow-growth threshold (the size at which the rate of increase is slowed) to avoid congestion on the network again.<br><br>• The rightmost number (31) is the maximum value currently allowed for the VC based on channel limitations.<br><br>**Reference:** See Appendix G for PEDRIVER congestion control and channel selection information. |
| ❾ Pipe Quota Reached | Indicates the number of times the entire transmit window was full. If this number is small as compared with the number of sequenced messages transmitted, it indicates that the local node is not sending large bursts of data to the remote node. |
| ❿ Xmt C/T (transmission count/target) | Shows both the number of successful transmissions since the last time the pipe quota was increased and the target value at which the pipe quota is allowed to increase. In the example, the count is 0 because the pipe quota is already at its maximum value (31), so successful transmissions are not being counted. |
| ⓫ RndTrp uS (round trip in microseconds) | Displays values that are used to calculate the retransmission timeout in microseconds. The leftmost number (18540) is the average round-trip time, and the rightmost number (7764) is the average variation in round-trip time. In the example, the values indicate that the round trip is about 19 milliseconds plus or minus about 8 milliseconds. |

(continued on next page)

**Table F–5 (Cont.) SHOW PORT/VC Display**

| Field | Description |
|---|---|
| ⓬ Open and Cls | Displays open (Open) and closed (Cls) timestamps for the last significant changes in the virtual circuit. The repeated loss of one or more virtual circuits over a short period of time (fewer than 10 minutes) indicates network problems. |
| ⓭ Cls | If you are analyzing a crash dump, you should check whether the crash-dump time corresponds to the timestamp for channel closures (Cls). |

## F.3.4 Monitoring PEDRIVER Buses

The SDA command SHOW PORT/BUS=BUS_*LAN-device* command is useful for displaying the PEDRIVER representation of a LAN adapter. To PEDRIVER, a bus is the logical representation of the LAN adapter. (To list the names and addresses of buses, enter the SDA command SHOW PORT/ADDR=PE_PDT and then press the Return key twice.) Example F–3 shows a display for the LAN adapter named EXA.

**Example F–3 SDA Command SHOW PORT/BUS Display**

```
SDA> SHOW PORT/BUS=BUS_EXA
VAXcluster data structures
--------------------------
--- BUS: 817E02C0  (EXA)  Device: EX_DEMNA  LAN Address: AA-00-04-00-64-4F ---
                                 LAN Hardware Address: 08-00-2B-2C-20-B5
Status: 00000803 run,online❶,restart
------- Transmit ------  ------- Receive -------  ---- Structure Addresses ---
Msg Xmt        20290620  Msg Rcv        67321527  PORT Address        817E1140
  Mcast Msgs    1318437    Mcast Msgs   39773666  VCIB Addr           817E0478
  Mcast Bytes 168759936    Mcast Bytes 159660184  HELLO Message Addr  817E0508
Bytes Xmt    2821823510  Bytes Rcv    3313602089  BYE Message Addr    817E0698
Outstand I/Os         0  Buffer Size        1424  Delete BUS Rtn Adr  80C6DA46
Xmt Errors❷       15896  Rcv Ring Size        31
Last Xmt Error 0000005C        Time of Last Xmt Error❸21-JAN-1994 15:33:38.96
--- Receive Errors ----  ------ BUS Timer ------  ----- Datalink Events ------
TR Mcast Rcv          0  Handshake TMO  80C6F070  Last  7-DEC-1992 17:15:42.18
Rcv Bad SCSID         0  Listen TMO     80C6F074  Last Event          00001202
Rcv Short Msg         0  HELLO timer           3  Port Usable                1
Fail CH Alloc         0  HELLO Xmt err❹     1623  Port Unusable              0
Fail VC Alloc         0                          Address Change             1
Wrong PORT            0                          Port Restart Fail          0
```

| Field | Description |
|---|---|
| ❶ Status: | The Status line should always display a status of "online" to indicate that PEDRIVER can access its LAN adapter. |
| ❷ Xmt Errors (transmission errors) | Indicates the number of times PEDRIVER has been unable to transmit a packet using this LAN adapter. |
| ❸ Time of Last Xmt Error | You can compare the time shown in this field with the Open and Cls times shown in the VC display in Example F–2 to determine whether the time of the LAN adapter failure is close to the time of a virtual circuit failure.<br>**Note:** Transmission errors at the LAN adapter bus level cause a virtual circuit breakage. |

| Field | Description |
|---|---|
| ❹ HELLO Xmt err (HELLO transmission error) | Indicates how many times a message transmission failure has "dropped" a PEDRIVER HELLO datagram message. (The Channel Control [CC] level description in Section F.1 briefly describes the purpose of HELLO datagram messages.) If many HELLO transmission errors occur, PEDRIVER on other nodes probably is timing out a channel, which could eventually result in closure of the virtual circuit.<br><br>The 1623 HELLO transmission failures shown in Example F–3 contributed to the high number of transmission errors (15896). Note that it is impossible to have a low number of transmission errors and a high number of HELLO transmission errors. |

## F.3.5 Monitoring LAN Adapters

Use the SDA command SHOW LAN/COUNT to display information about the LAN adapters as maintained by the LAN device driver (the command shows counters for all protocols, not just PEDRIVER [SCA] related counters). Example F–4 shows a sample display from the SHOW LAN/COUNT command.

### Example F–4 SDA Command SHOW LAN/COUNTERS Display

```
$ ANALYZE/SYSTEM
SDA> SHOW LAN/COUNTERS

LAN Data Structures
-------------------
            -- EXA Counters Information 22-JAN-1994 11:21:19 --

Seconds since zeroed         3953329    Station failures                   0
Octets received          13962888501    Octets sent               11978817384
PDUs received              121899287    PDUs sent                    76872280
Mcast octets received     7494809802    Mcast octets sent           183142023
Mcast PDUs received         58046934    Mcast PDUs sent               1658028
Unrec indiv dest PDUs              0    PDUs sent, deferred           4608431
Unrec mcast dest PDUs              0    PDUs sent, one coll           3099649
Data overruns                      2    PDUs sent, mul coll           2439257
Unavail station buffs❶             0    Excessive collisions❷            5059
Unavail user buffers               0    Carrier check failure               0
Frame check errors               483    Short circuit failure               0
Alignment errors               10215    Open circuit failure                0
Frames too long                  142    Transmits too long                  0
Rcv data length error              0    Late collisions                 14931
802E PDUs received             28546    Coll detect chk fail                0
802 PDUs received                  0    Send data length err                0
Eth PDUs received          122691742    Frame size errors                   0

LAN Data Structures
-------------------
        -- EXA Internal Counters Information 22-JAN-1994 11:22:28 --

Internal counters address   80C58257    Internal counters size             24
Number of ports                    0    Global page transmits               0
No work transmits            3303771    SVAPTE/BOFF transmits               0
Bad PTE transmits                  0    Buffer_Adr transmits                0

Fatal error count                  0    RDL errors                          0
Transmit timeouts                  0    Last fatal error                 None
Restart failures                   0    Prev fatal error                 None
Power failures                     0    Last error CSR               00000000
Hardware errors                    0    Fatal error code                 None
Control timeouts                   0    Prev fatal error                 None

Loopback sent                      0    Loopback failures                   0
System ID sent                     0    System ID failures                  0
ReqCounters sent                   0    ReqCounters failures                0
```

(continued on next page)

**Example F–4 (Cont.) SDA Command SHOW LAN/COUNTERS Display**

```
      -- EXA1 60-07 (SCA) Counters Information 22-JAN-1994 11:22:31 --

Last receive❸       22-JAN 11:22:31    Last transmit❸    22-JAN 11:22:31
Octets received          7616615830    Octets sent              2828248622
PDUs received              67375315    PDUs sent                  20331888
Mcast octets received             0    Mcast octets sent                 0
Mcast PDUs received               0    Mcast PDUs sent                   0
Unavail user buffer               0    Last start attempt             None
Last start done      7-DEC 17:12:29    Last start failed              None
   .
   .
   .
```

The SHOW LAN/COUNTERS display usually includes device counter information about several LAN adapters. However, for purposes of example, only one device is shown in Example F–4.

| Field | Description |
|---|---|
| ❶ Unavail station buffs (unavailable station buffers) | Records the number of times that fixed station buffers in the LAN driver were unavailable for incoming packets. The node receiving a message can lose packets when the node does not have enough LAN station buffers. (LAN buffers are used by a number of consumers other than PEDRIVER, such as DECnet, TCP/IP, and LAT.) Packet loss because of insufficient LAN station buffers is a symptom of either LAN adapter congestion or the system's inability to reuse the existing buffers fast enough. |
| ❷ Excessive collisions | Indicates the number of unsuccessful attempts to transmit messages on the adapter. This problem is often caused by:<br>• A LAN loading problem resulting from heavy traffic (70% to 80% utilization) on the specific LAN segment.<br>• A component called a screamer. A **screamer** is an adapter whose protocol does not adhere to Ethernet or FDDI hardware protocols. A screamer does not wait for permission to transmit packets on the adapter, thereby causing collision errors to register in this field.<br>If a significant number of transmissions with multiple collisions have occurred, then OpenVMS Cluster performance is degraded. You might be able to improve performance either by removing some nodes from the LAN segment or by adding another LAN segment to the cluster. The overall goal is to reduce traffic on the existing LAN segment, thereby making more bandwidth available to the OpenVMS Cluster system. |
| ❸ Last receive and Last transmit | The difference in the times shown in the Last receive and Last transmit message fields should not be large. Minimally, the timestamps in these fields should reflect that HELLO datagram messages are being sent across channels every 3 seconds. Large time differences might indicate:<br>– A hardware failure<br>– Whether or not the LAN driver sees the NISCA protocol as being active on a specific LAN adapter |

# F.4 Troubleshooting NISCA Communications

## F.4.1 Areas of Trouble

Sections F.5 and F.6 describe two likely areas of trouble for LAN networks: channel formation and retransmission. The discussions of these two problems often include references to the use of a LAN analyzer tool to isolate information in the NISCA protocol.

**Reference:** As you read about how to diagnose NISCA problems, you may also find it helpful to refer to Section F.7, which describes the NISCA protocol packet, and Section F.8, which describes how to choose and use a LAN network failure analyzer.

# F.5 Channel Formation

Channel-formation problems occur when two nodes cannot communicate properly between LAN adapters.

## F.5.1 How Channels Are Formed

Table F–6 provides a step-by-step description of channel formation.

**Table F–6 Channel Formation**

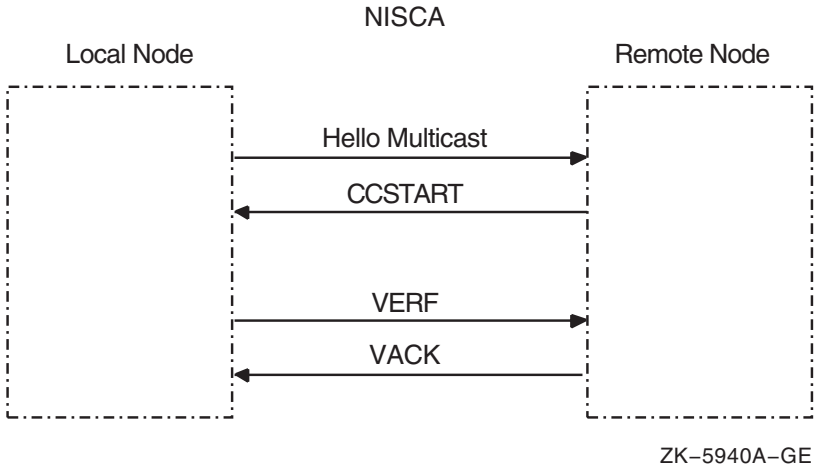
| Step | Action |
|---|---|
| 1 | Channels are formed when a node sends a HELLO datagram from its LAN adapter to a LAN adapter on another cluster node. If this is a new remote LAN adapter address, or if the corresponding channel is closed, the remote node receiving the HELLO datagram sends a CCSTART datagram to the originating node after a delay of up to 2 seconds. |
| 2 | Upon receiving a CCSTART datagram, the originating node verifies the cluster password and, if the password is correct, the node responds with a VERF datagram and waits for up to 5 seconds for the remote node to send a VACK datagram. (VERF, VACK, CCSTART, and HELLO datagrams are described in Section F.7.6.) |
| 3 | Upon receiving a VERF datagram, the remote node verifies the cluster password; if the password is correct, the node responds with a VACK datagram and marks the channel as open. (See Figure F–2.) |
| 4 | |

| WHEN the local node... | THEN... |
|---|---|
| Does not receive the VACK datagram within 5 seconds | The channel state goes back to closed and the handshake timeout counter is incremented. |
| Receives the VACK datagram within 5 seconds and the cluster password is correct | The channel is opened. |

| | |
|---|---|
| 5 | Once a channel has been formed, it is maintained (kept open) by the regular multicast of HELLO datagram messages. Each node multicasts a HELLO datagram message at least once every 3.0 seconds over each LAN adapter. Either of the nodes sharing a channel closes the channel with a listen timeout if it does not receive a HELLO datagram or a sequence message from the other node within 8 to 9 seconds. If you receive a “Port closed virtual circuit” message, it indicates a channel was formed but there is a problem receiving traffic on time. When this happens, look for HELLO datagram messages getting lost. |

Figure F–2 shows a message exchange during a successful channel-formation handshake.

**Figure F–2 Channel-Formation Handshake**




## F.5.2 Techniques for Troubleshooting

When there is a break in communications between two nodes and you suspect problems with channel formation, follow these instructions:

| Step | Action |
|---|---|
| **1** | Check the obvious:<br>• Is the remote node powered on?<br>• Is the remote node booted?<br>• Are the required network connections connected?<br>• Do the cluster multicast datagrams pass through all of the required bridges in both directions?<br>• Are the cluster group code and password values the same on all nodes? |
| **2** | Check for dead channels by using SDA. The SDA command SHOW PORT/CHANNEL/VC=VC_*remote_node* can help you determine whether a channel ever existed; the command displays the channel's state.<br>**Reference:** Refer to Section F.3 for examples of the SHOW PORT command. Section F.10.1 describes how to use a LAN analyzer to troubleshoot channel formation problems. |
| **3** | See also Appendix D for information about using the LAVC$FAILURE_ANALYSIS program to troubleshoot channel problems. |

# F.6 Retransmission Problems

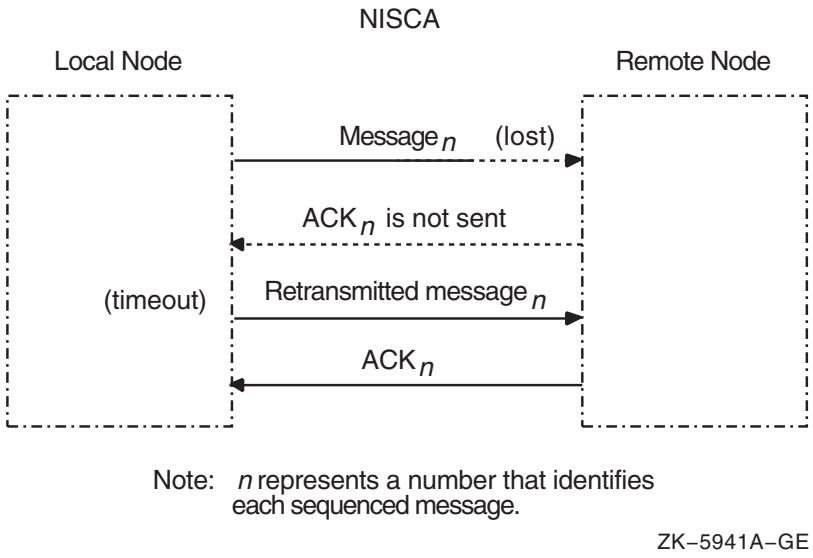
Retransmissions occur when the local node does not receive acknowledgment of a message in a timely manner.

## F.6.1 Why Retransmissions Occur

The first time the sending node transmits the datagram containing the sequenced message data, PEDRIVER sets the value of the REXMT flag bit in the TR header to 0. If the datagram requires retransmission, PEDRIVER sets the REXMT flag bit to 1 and resends the datagram. PEDRIVER retransmits the datagram until either the datagram is received or the virtual circuit is closed. If multiple channels are available, PEDRIVER attempts to retransmit the message on a different channel in an attempt to avoid the problem that caused the retransmission.

Retransmission typically occurs when a node runs out of a critical resource, such as large request packets (LRPs) or nonpaged pool, and a message is lost after it reaches the remote node. Other potential causes of retransmissions include overloaded LAN bridges, slow LAN adapters (such as the DELQA), and heavily loaded systems, which delay packet transmission or reception. Figure F–3 shows an unsuccessful transmission followed by a successful retransmission.

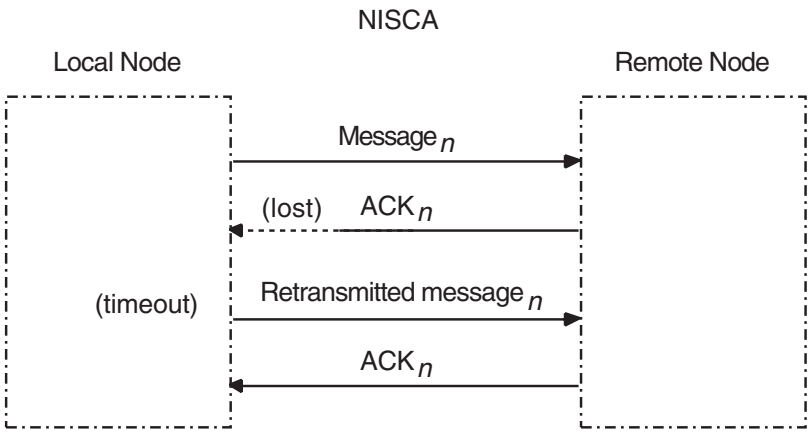
**Figure F–3 Lost Messages Cause Retransmissions**




Because the first message was lost, the local node does not receive acknowledgment (ACK) from the remote node. The remote node acknowledged the second (successful) transmission of the message.

Retransmission can also occur if the cables are seated improperly, if the network is too busy and the datagram cannot be sent, or if the datagram is corrupted or lost during transmission either by the originating LAN adapter or by any bridges or repeaters. Figure F–4 illustrates another type of retransmission.

**Figure F–4 Lost ACKs Cause Retransmissions**




In Figure F–4, the remote node receives the message and transmits an acknowledgment (ACK) to the sending node. However, because the ACK from the receiving node is lost, the sending node retransmits the message.

### F.6.2 Techniques for Troubleshooting

You can troubleshoot cluster retransmissions using a LAN protocol analyzer for each LAN segment. If multiple segments are used for cluster communications, then the LAN analyzers need to support a distributed enable and trigger mechanism (see Section F.8). See also Section G.1 for more information about how PEDRIVER chooses channels on which to transmit datagrams.

**Reference:** Techniques for isolating the retransmitted datagram using a LAN analyzer are discussed in Section F.10.2. See also Appendix G for more information about congestion control and PEDRIVER message retransmission.

## F.7 Understanding NISCA Datagrams

Troubleshooting NISCA protocol communication problems requires an understanding of the NISCA protocol packet that is exchanged across the OpenVMS Cluster system.

### F.7.1 Packet Format

The format of packets on the NISCA protocol is defined by the $NISCADEF macro, which is located in [DRIVER.LIS] on VAX systems and in [LIB.LIS] for Alpha systems on your CD listing disk.

Figure F–5 shows the general form of NISCA datagrams. A NISCA datagram consists of the following headers, which are usually followed by user data:

- LAN headers, including an Ethernet or an FDDI header
- Datagram exchange (DX) header

- Channel control (CC) or transport (TR) header

**Figure F–5 NISCA Headers**


**Caution:** The NISCA protocol is subject to change without notice.

## F.7.2 LAN Headers

The NISCA protocol is supported on LANs consisting of Ethernet and FDDI, described in Sections F.7.3 and F.7.4. These headers contain information that is useful for diagnosing problems that occur between LAN adapters.

**Reference:** See Section F.9.4 for methods of isolating information in LAN headers.

## F.7.3 Ethernet Header

Each datagram that is transmitted or received on the Ethernet is prefixed with an Ethernet header. The Ethernet header, shown in Figure F–6 and described in Table F–7, is 16 bytes long.

**Figure F–6 Ethernet Header**

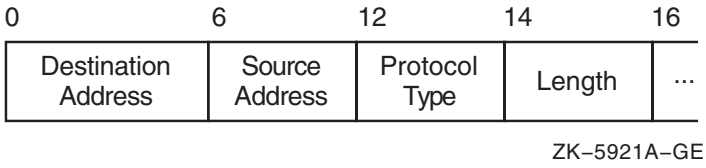



**Table F–7 Fields in the Ethernet Header**

| Field | Description |
|---|---|
| Destination address | LAN address of the adapter that should receive the datagram |
| Source address | LAN address of the adapter sending the datagram |
| Protocol type | NISCA protocol (60–07) hexadecimal |
| Length | Number of data bytes in the datagram following the length field |

## F.7.4 FDDI Header

Each datagram that is transmitted or received on the FDDI is prefixed with an FDDI header. The NISCA protocol uses mapped Ethernet format datagrams on the FDDI. The FDDI header, shown in Figure F–7 and described in Table F–8, is 23 bytes long.

**Figure F–7 FDDI Header**

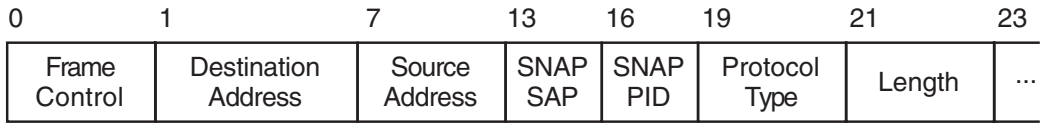



**Table F–8 Fields in the FDDI Header**

| Field | Description |
|---|---|
| Frame control | NISCA datagrams are logical link control (LLC) frames with a priority value (5*x*). The low-order 3 bits of the frame-control byte contain the priority value. All NISCA frames are transmitted with a nonzero priority field. Frames received with a zero-priority field are assumed to have traveled over an Ethernet segment because Ethernet packets do not have a priority value and because Ethernet-to-FDDI bridges generate a priority value of 0. |
| Destination address | LAN address of the adapter that should receive the datagram. |
| Source address | LAN address of the adapter sending the datagram. |
| SNAP SAP | Subnetwork access protocol; service access point. The value of the access point is AA–AA–03 hexadecimal. |
| SNAP PID | Subnetwork access protocol; protocol identifier. The value of the identifier is 00–00–00 hexadecimal. |
| Protocol type | NISCA protocol (60–07) hexadecimal. |
| Length | Number of data bytes in the datagram following the length field. |

## F.7.5 Datagram Exchange (DX) Header

The datagram exchange (DX) header for the OpenVMS Cluster protocol is used to address the data to the correct OpenVMS Cluster node. The DX header, shown in Figure F–8 and described in Table F–9, is 14 bytes long. It contains information that describes the OpenVMS Cluster connection between two nodes. See Section F.9.3 about methods of isolating data for the DX header.

**Figure F–8 DX Header**

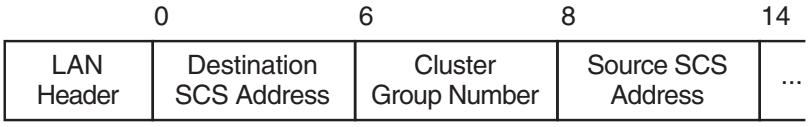


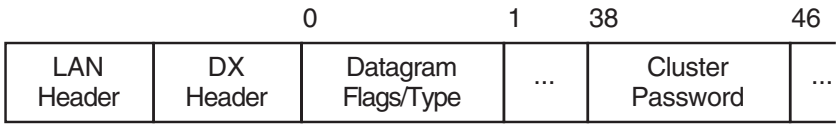
**Table F–9 Fields in the DX Header**

| Field | Description |
|---|---|
| Destination SCS address | Manufactured using the address AA–00–04–00–*remote-node*-SCSSYSTEMID. Append the remote node's SCSSYSTEMID system parameter value for the low-order 16 bits. This address represents the destination SCS transport address or the OpenVMS Cluster multicast address. |
| Cluster group number | The cluster group number specified by the system manager. See Chapter 8 for more information about cluster group numbers. |
| Source SCS address | Represents the source SCS transport address and is manufactured using the address AA–00–04–00–*local-node*-SCSSYSTEMID. Append the local node's SCSSYSTEMID system parameter value as the low-order 16 bits. |

## F.7.6 Channel Control (CC) Header

The channel control (CC) message is used to form and maintain working network paths between nodes in the OpenVMS Cluster system. The important fields for network troubleshooting are the datagram flags/type and the cluster password. Note that because the CC and TR headers occupy the same space, there is a TR/CC flag that identifies the type of message being transmitted over the channel. Figure F–9 shows the portions of the CC header needed for network troubleshooting, and Table F–10 describes these fields.

**Figure F–9 CC Header**




ZK–5924A–GE

**Table F–10 Fields in the CC Header**

| Field | Description |
|---|---|
| Datagram type (bits <3:0>) | Identifies the type of message on the Channel Control level. The following table shows the datagrams and their functions. |

| Value | Abbreviated Datagram Type | Expanded Datagram Type | Function |
|---|---|---|---|
| 0 | HELLO | HELLO datagram message | Multicast datagram that initiates the formation of a channel between cluster nodes and tests and maintains the existing channels. This datagram does not contain a valid cluster password. |
| 1 | BYE | Node-stop notification | Datagram that signals the departure of a cluster node. |
| 2 | CCSTART | Channel start | Datagram that starts the channel-formation handshake between two cluster nodes. This datagram is sent in response to receiving a HELLO datagram from an unknown LAN adapter address. |
| 3 | VERF | Verify | Datagram that acknowledges the CCSTART datagram and continues the channel formation handshake. The datagram is sent in response to receiving a CCSTART or SOLICIT_SRV datagram. |
| 4 | VACK | Verify acknowledge | Datagram that completes the channel-formation handshake. The datagram is sent in response to receiving a VERF datagram. |
| 5 | Reserved | | |
| 6 | SOLICIT_SERVICE | Solicit | Datagram sent by a booting node to form a channel to its disk server. The server responds by sending a VERF, which forms the channel. |
| 7–15 | Reserved | | |

| | |
|---|---|
| Datagram flags (bits <7:4>) | Provide additional information about the control datagram. The following bits are defined:<br>• Bit <4> (AUTHORIZE)—Set to 1 if the cluster password field is valid.<br>• Bit <5> (Reserved)—Set to 1.<br>• Bit <6> (Reserved)—Set to 0.<br>• Bit <7> (TR/CC flag)—Set to 1 to indicate the CC datagram. |
| Cluster password | Contains the cluster password. |

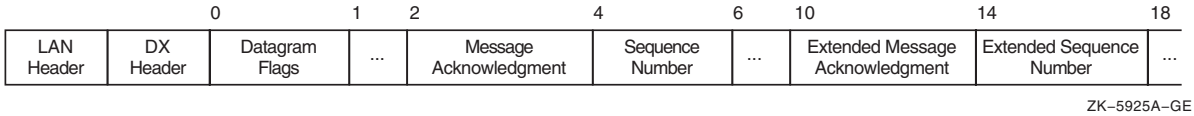
## F.7.7 Transport (TR) Header

The transport (TR) header is used to pass SCS datagrams and sequenced messages between cluster nodes. The important fields for network troubleshooting are the TR datagram flags, message acknowledgment, and sequence numbers. Note that because the CC and TR headers occupy the same

space, a TR/CC flag identifies the type of message being transmitted over the channel.

Figure F–10 shows the portions of the TR header that are needed for network troubleshooting, and Table F–11 describes these fields.

**Figure F–10 TR Header**




**Note:** The TR header shown in Figure F–10 is used when both nodes are running Version 1.4 or later of the NISCA protocol. If one or both nodes are running Version 1.3 or an earlier version of the protocol, then both nodes will use the message acknowledgment and sequence number fields in place of the extended message acknowledgment and extended sequence number fields, respectively.

**Table F–11 Fields in the TR Header**

| Field | Description | | | |
|---|---|---|---|---|
| Datagram flags (bits <7:0>) | Provide additional information about the transport datagram. | | | |
| | **Value** | **Abbreviated Datagram Type** | **Expanded Datagram Type** | **Function** |
| | 0 | DATA | Packet data | Contains data to be delivered to the upper levels of software. |
| | 1 | SEQ | Sequence flag | Set to 1 if this is a sequenced message and the sequence number is valid. |
| | 2 | Reserved | | Set to 0. |
| | 3 | ACK | Acknowledgment | Acknowledges the field is valid. |
| | 4 | RSVP | Reply flag | Set when an ACK datagram is needed immediately. |
| | 5 | REXMT | Retransmission | Set for all retransmissions of a sequenced message. |
| | 6 | Reserved | | Set to 0. |
| | 7 | TR/CC flag | Transport flag | Set to 0; indicates a TR datagram. |
| Message acknowledgment | An increasing value that specifies the last sequenced message segment received by the local node. All messages prior to this value are also acknowledged. This field is used when one or both nodes are running Version 1.3 or earlier of the NISCA protocol. | | | |
| Extended message acknowledgment | An increasing value that specifies the last sequenced message segment received by the local node. All messages prior to this value are also acknowledged. This field is used when both nodes are running Version 1.4 or later of the NISCA protocol. | | | |

(continued on next page)

**Table F–11 (Cont.) Fields in the TR Header**

| Field | Description |
|---|---|
| Sequence number | An increasing value that specifies the order of datagram transmission from the local node. This number is used to provide guaranteed delivery of this sequenced message segment to the remote node. This field is used when one or both nodes are running Version 1.3 or earlier of the NISCA protocol. |
| Extended sequence number | An increasing value that specifies the order of datagram transmission from the local node. This number is used to provide guaranteed delivery of this sequenced message segment to the remote node. This field is used when both nodes are running Version 1.4 or later of the NISCA protocol. |

## F.8 Using a LAN Protocol Analysis Program

Some failures, such as packet loss resulting from congestion, intermittent network interruptions of less than 20 seconds, problems with backup bridges, and intermittent performance problems, can be difficult to diagnose. Intermittent failures may require the use of a LAN analysis tool to isolate and troubleshoot the NISCA protocol levels described in Section F.1.

As you evaluate the various network analysis tools currently available, you should look for certain capabilities when comparing LAN analyzers. The following sections describe the required capabilities.

### F.8.1 Single or Multiple LAN Segments

Whether you need to troubleshoot problems on a single LAN segment or on multiple LAN segments, a LAN analyzer should help you isolate specific patterns of data. Choose a LAN analyzer that can isolate data matching unique patterns that you define. You should be able to define data patterns located in the data regions following the LAN header (described in Section F.7.2). In order to troubleshoot the NISCA protocol properly, a LAN analyzer should be able to match multiple data patterns simultaneously.

To troubleshoot single or multiple LAN segments, you must minimally define and isolate transmitted and retransmitted data in the TR header (see Section F.7.7). Additionally, for effective network troubleshooting across multiple LAN segments, a LAN analysis tool should include the following functions:

- A **distributed enable** function that allows you to synchronize multiple LAN analyzers that are set up at different locations so that they can capture information about the same event as it travels through the LAN configuration

- A **distributed combination trigger** function that automatically triggers multiple LAN analyzers at different locations so that they can capture information about the same event

The purpose of distributed enable and distributed combination trigger functions is to capture packets as they travel across multiple LAN segments. The implementation of these functions discussed in the following sections use multicast messages to reach all LAN segments of the extended LAN in the system configuration. By providing the ability to synchronize several LAN analyzers at different locations across multiple LAN segments, the distributed enable and combination trigger functions allow you to troubleshoot LAN configurations that span multiple sites over several miles.

### F.8.2 Multiple LAN Segments

To troubleshoot multiple LAN segments, LAN analyzers must be able to capture the multicast packets and dynamically enable the trigger function of the LAN analyzer, as follows:

| Step | Action |
|---|---|
| **1** | Start capturing the data according to the rules specific to your LAN analyzer. Compaq recommends that only one LAN analyzer transmit a distributed enable multicast packet on the LAN. The packet must be transmitted according to the media access-control rules. |
| **2** | Wait for the distributed enable multicast packet. When the packet is received, enable the distributed combination trigger function. Prior to receiving the distributed enable packet, all LAN analyzers must be able to ignore the trigger condition. This feature is required in order to set up multiple LAN analyzers capable of capturing the same event. Note that the LAN analyzer transmitting the distributed enable should not wait to receive it. |
| **3** | Wait for an explicit (user-defined) trigger event or a distributed trigger packet. When the LAN analyzer receives either of these triggers, the LAN analyzer should stop the data capture.<br>Prior to receiving either trigger, the LAN analyzer should continue to capture the requested data. This feature is required in order to allow multiple LAN analyzers to capture the same event. |
| **4** | Once triggered, the LAN analyzer completes the distributed trigger function to stop the other LAN analyzers from capturing data related to the event that has already occurred. |

The HP 4972A LAN Protocol Analyzer, available from the Hewlett-Packard Company, is one example of a network failure analysis tool that provides the required functions described in this section.

**Reference:** Section F.10 provides examples that use the HP 4972A LAN Protocol Analyzer.

## F.9 Data Isolation Techniques

The following sections describe the types of data you should isolate when you use a LAN analysis tool to capture OpenVMS Cluster data between nodes and LAN adapters.

### F.9.1 All OpenVMS Cluster Traffic

To isolate all OpenVMS Cluster traffic on a specific LAN segment, capture all the packets whose LAN header contains the protocol type 60–07.

**Reference:** See also Section F.7.2 for a description of the LAN headers.

### F.9.2 Specific OpenVMS Cluster Traffic

To isolate OpenVMS Cluster traffic for a specific cluster on a specific LAN segment, capture packets in which:

- The LAN header contains the the protocol type 60–07.
- The DX header contains the cluster group number specific to that OpenVMS Cluster.

**Reference:** See Sections F.7.2 and F.7.5 for descriptions of the LAN and DX headers.

### F.9.3 Virtual Circuit (Node-to-Node) Traffic

To isolate virtual circuit traffic between a specific pair of nodes, capture packets in which the LAN header contains:

- The protocol type 60–07
- The destination SCS address
- The source SCS address

You can further isolate virtual circuit traffic between a specific pair of nodes to a specific LAN segment by capturing the following additional information from the DX header:

- The cluster group code specific to that OpenVMS Cluster
- The destination SCS transport address
- The source SCS transport address

**Reference:** See Sections F.7.2 and F.7.5 for LAN and DX header information.

### F.9.4 Channel (LAN Adapter–to–LAN Adapter) Traffic

To isolate channel information, capture all packet information on every channel between LAN adapters. The DX header contains information useful for diagnosing heavy communication traffic between a pair of LAN adapters. Capture packets in which the LAN header contains:

- The destination LAN adapter address
- The source LAN adapter address

Because nodes can use multiple LAN adapters, specifying the source and destination LAN addresses may not capture all of the traffic for the node. Therefore, you must specify a channel as the source LAN address and the destination LAN address in order to isolate traffic on a specific channel.

**Reference:** See Section F.7.2 for information about the LAN header.

### F.9.5 Channel Control Traffic

To isolate channel control traffic, capture packets in which:

- The LAN header contains the the protocol type 60–07.
- The CC header datagram flags byte (the TR/CC flag, bit <7>) is set to 1.

**Reference:** See Sections F.7.2 and F.7.6 for a description of the LAN and CC headers.

### F.9.6 Transport Data

To isolate transport data, capture packets in which:

- The LAN header contains the the protocol type 60–07.
- The TR header datagram flags byte (the TR/CC flag, bit <7>) is set to 0.

**Reference:** See Sections F.7.2 and F.7.7 for a description of the LAN and TR headers.

# F.10 Setting Up an HP 4972A LAN Protocol Analyzer

The HP 4972A LAN Protocol Analyzer, available from the Hewlett-Packard Company, is highlighted here because it meets all of the requirements listed in Section F.8. However, the HP 4972A LAN Protocol Analyzer is merely representative of the type of product useful for LAN network troubleshooting.

**Note:** Use of this particular product as an example here should not be construed as a specific purchase requirement or endorsement.

This section provides some examples of how to set up the HP 4972A LAN Protocol Analyzer to troubleshoot the local area OpenVMS Cluster system protocol for channel formation and retransmission problems.

## F.10.1 Analyzing Channel Formation Problems

If you have a LAN protocol analyzer, you can set up filters to capture data related to the channel control header (described in Section F.7.6).

You can trigger the LAN analyzer by using the following datagram fields:

- Protocol type set to 60–07 hexadecimal
- Correct cluster group number
- TR/CC flag set to 1

Then look for the HELLO, CCSTART, VERF, and VACK datagrams in the captured data. The CCSTART, VERF, VACK, and SOLICIT_SRV datagrams should have the AUTHORIZE bit (bit <4>) set in the CC flags byte. Additionally, these messages should contain the scrambled cluster password (nonzero authorization field). You can find the scrambled cluster password and the cluster group number in the first four longwords of SYS$SYSTEM:CLUSTER_AUTHORIZE.DAT file.

**Reference:** See Sections F.9.3 through F.9.5 for additional data isolation techniques.

## F.10.2 Analyzing Retransmission Problems

Using a LAN analyzer, you can trace datagrams as they travel across an OpenVMS Cluster system, as described in Table F–12.

**Table F–12 Tracing Datagrams**

| Step | Action |
|---|---|
| 1 | Trigger the analyzer using the following datagram fields:<br>• Protocol type set to 60–07<br>• Correct cluster group number<br>• TR/CC flag set to 0<br>• REXMT flag set to 1 |

(continued on next page)

**Table F–12 (Cont.) Tracing Datagrams**

| Step | Action |
|---|---|
| **2** | Use the distributed enable function to allow the same event to be captured by several LAN analyzers at different locations. The LAN analyzers should start the data capture, wait for the distributed enable message, and then wait for the explicit trigger event or the distributed trigger message. Once triggered, the analyzer should complete the distributed trigger function to stop the other LAN analyzers capturing data. |
| **3** | Once all the data is captured, locate the sequence number (for nodes running the NISCA protocol Version 1.3 or earlier) or the extended sequence number (for nodes running the NISCA protocol Version 1.4 or later) for the datagram being retransmitted (the datagram with the REXMT flag set). Then, search through the previously captured data for another datagram between the same two nodes (not necessarily the same LAN adapters) with the following characteristics:<br>• Protocol type set to 60–07<br>• Same DX header as the datagram with the REXMT flag set<br>• TR/CC flag set to 0<br>• REXMT flag set to 0<br>• Same sequence number or extended sequence number as the datagram with the REXMT flag set |

(continued on next page)

**Table F–12 (Cont.) Tracing Datagrams**

| Step | Action |
|---|---|
| 4 | The following techniques provide a way of searching for the problem's origin. |

| IF... | THEN... |
|---|---|
| The datagram appears to be corrupt | Use the LAN analyzer to search in the direction of the source node for the corruption cause. |
| The datagram appears to be correct | Search in the direction of the destination node to ensure that the datagram gets to its destination. |
| The datagram arrives successfully at its LAN segment destination | Look for a TR packet from the destination node containing the sequence number (for nodes running the NISCA protocol Version 1.3 or earlier) or the extended sequence number (for nodes running the NISCA protocol Version 1.4 or later) in the message acknowledgment or extended message acknowledgement field. ACK datagrams have the following fields set:<br>• Protocol type set to 60–07<br>• Same DX header as the datagram with the REXMT flag set<br>• TR/CC flag set to 0<br>• ACK flag set to 1 |
| The acknowledgment was not sent, or if a significant delay occurred between the reception of the message and the transmission of the acknowledgment | Look for a problem with the destination node and LAN adapter. Then follow the ACK packet through the network. |
| The ACK arrives back at the node that sent the retransmission packet | Either of the following conditions may exist:<br>• The retransmitting node is having trouble receiving LAN data.<br>• The round-trip delay of the original datagram exceeded the estimated timeout value.<br>You can verify the second possibility by using SDA and looking at the ReRcv field of the virtual circuit display of the system receiving the retransmitted datagram.<br>**Reference:** See Example F–2 for an example of this type of SDA display. |

**Reference:** See Appendix G for more information about congestion control and PEDRIVER message retransmission.

## F.11 Filters

This section describes:

- How to use the HP 4972A LAN Protocol Analyzer filters to isolate packets that have been retransmitted or that are specific to a particular OpenVMS Cluster.
- How to enable the distributed enable and trigger functions.

### F.11.1 Capturing All LAN Retransmissions for a Specific OpenVMS Cluster

Use the values shown in Table F–13 to set up a filter, named LAVc_TR_ReXMT, for all of the LAN retransmissions for a specific cluster. Fill in the value for the local area OpenVMS Cluster group code (*nn–nn*) to isolate a specific OpenVMS Cluster on the LAN.

**Table F–13 Capturing Retransmissions on the LAN**

| Byte Number | Field | Value |
|---|---|---|
| 1 | DESTINATION | *xx–xx–xx–xx–xx–xx* |
| 7 | SOURCE | *xx–xx–xx–xx–xx–xx* |
| 13 | TYPE | 60–07 |
| 23 | LAVC_GROUP_CODE | *nn–nn* |
| 31 | TR FLAGS | $0x1xxxxx_2$ |
| 33 | ACKING MESSAGE | *xx–xx* |
| 35 | SENDING MESSAGE | *xx–xx* |

### F.11.2 Capturing All LAN Packets for a Specific OpenVMS Cluster

Use the values shown in Table F–14 to filter all of the LAN packets for a specific cluster. Fill in the value for OpenVMS Cluster group code (*nn–nn*) to isolate a specific OpenVMS Cluster on the LAN. The filter is named LAVc_all.

**Table F–14 Capturing All LAN Packets (LAVc_all)**

| Byte Number | Field | Value |
|---|---|---|
| 1 | DESTINATION | *xx–xx–xx–xx–xx–xx* |
| 7 | SOURCE | *xx–xx–xx–xx–xx–xx* |
| 13 | TYPE | 60–07 |
| 23 | LAVC_GROUP_CODE | *nn–nn* |
| 33 | ACKING MESSAGE | *xx–xx* |
| 35 | SENDING MESSAGE | *xx–xx* |

### F.11.3 Setting Up the Distributed Enable Filter

Use the values shown in Table F–15 to set up a filter, named Distrib_Enable, for the distributed enable packet received event. Use this filter to troubleshoot multiple LAN segments.

**Table F–15 Setting Up a Distributed Enable Filter (Distrib_Enable)**

| Byte Number | Field | Value | ASCII |
|---|---|---|---|
| 1 | DESTINATION | 01–4C–41–56–63–45 | .LAVcE |
| 7 | SOURCE | *xx–xx–xx–xx–xx–xx* | |

(continued on next page)

**Table F–15 (Cont.) Setting Up a Distributed Enable Filter (Distrib_Enable)**

| Byte Number | Field | Value | ASCII |
|---|---|---|---|
| 13 | TYPE | 60–07 | ‘. |
| 15 | TEXT | *xx* | |

### F.11.4 Setting Up the Distributed Trigger Filter

Use the values shown in Table F–16 to set up a filter, named Distrib_Trigger, for the distributed trigger packet received event. Use this filter to troubleshoot multiple LAN segments.

**Table F–16 Setting Up the Distributed Trigger Filter (Distrib_Trigger)**

| Byte Number | Field | Value | ASCII |
|---|---|---|---|
| 1 | DESTINATION | 01–4C–41–56–63–54 | .LAVcT |
| 7 | SOURCE | *xx–xx–xx–xx–xx–xx* | |
| 13 | TYPE | 60–07 | ‘. |
| 15 | TEXT | *xx* | |

## F.12 Messages

This section describes how to set up the distributed enable and distributed trigger messages.

### F.12.1 Distributed Enable Message

Table F–17 shows how to define the distributed enable message (Distrib_Enable) by creating a new message. You must replace the source address (*nn nn nn nn nn nn*) with the LAN address of the LAN analyzer.

**Table F–17 Setting Up the Distributed Enable Message (Distrib_Enable)**

| Field | Byte Number | Value | ASCII |
|---|---|---|---|
| Destination | 1 | 01 4C 41 56 63 45 | .LAVcE |
| Source | 7 | *nn nn nn nn nn nn* | |
| Protocol | 13 | 60 07 | ‘. |
| Text | 15 | 44 69 73 74 72 69 62 75 74 65 | Distribute |
| | 25 | 64 20 65 6E 61 62 6C 65 20 66 | d enable f |
| | 35 | 6F 72 20 74 72 6F 75 62 6C 65 | or trouble |
| | 45 | 73 68 6F 6F 74 69 6E 67 20 74 | shooting t |
| | 55 | 68 65 20 4C 6F 63 61 6C 20 41 | he Local A |
| | 65 | 72 65 61 20 56 4D 53 63 6C 75 | rea VMSclu |
| | 75 | 73 74 65 72 20 50 72 6F 74 6F | ster Proto |
| | 85 | 63 6F 6C 3A 20 4E 49 53 43 41 | col: NISCA |

### F.12.2 Distributed Trigger Message

Table F–18 shows how to define the distributed trigger message (Distrib_Trigger) by creating a new message. You must replace the source address (*nn nn nn nn nn nn*) with the LAN address of the LAN analyzer.

**Table F–18 Setting Up the Distributed Trigger Message (Distrib_Trigger)**

| Field | Byte Number | Value | ASCII |
|---|---|---|---|
| Destination | 1 | 01 4C 41 56 63 54 | .LAVcT |
| Source | 7 | *nn nn nn nn nn nn* | |
| Protocol | 13 | 60 07 | ‘. |
| Text | 15 | 44 69 73 74 72 69 62 75 74 65 | Distribute |
| | 25 | 64 20 74 72 69 67 67 65 72 20 | d trigger |
| | 35 | 66 6F 72 20 74 72 6F 75 62 6C | for troubl |
| | 45 | 65 73 68 6F 6F 74 69 6E 67 20 | eshooting |
| | 55 | 74 68 65 20 4C 6F 63 61 6C 20 | the Local |
| | 65 | 41 72 65 61 20 56 4D 53 63 6C | Area VMScl |
| | 75 | 75 73 74 65 72 20 50 72 6F 74 | uster Prot |
| | 85 | 6F 63 6F 6C 3A 20 4E 49 53 43 | ocol: NISC |
| | 95 | 41 | A |

## F.13 Programs That Capture Retransmission Errors

You can program the HP 4972 LAN Protocol Analyzer, as shown in the following source code, to capture retransmission errors. The starter program initiates the capture across all of the LAN analyzers. Only one LAN analyzer should run a copy of the starter program. Other LAN analyzers should run either the partner program or the scribe program. The partner program is used when the initial location of the error is unknown and when all analyzers should cooperate in the detection of the error. Use the scribe program to trigger on a specific LAN segment as well as to capture data from other LAN segments.

### F.13.1 Starter Program

The starter program initially sends the distributed enable signal to the other LAN analyzers. Next, this program captures all of the LAN traffic, and terminates as a result of either a retransmitted packet detected by this LAN analyzer or after receiving the distributed trigger sent from another LAN analyzer running the partner program.

The starter program shown in the following example is used to initiate data capture on multiple LAN segments using multiple LAN analyzers. The goal is to capture the data during the same time interval on all of the LAN segments so that the reason for the retransmission can be located.

```
Store: frames matching LAVc_all
 or Distrib_Enable
 or Distrib_Trigger
       ending with LAVc_TR_ReXMT
        or Distrib_Trigger

Log file: not used
```

```
Block 1:   Enable_the_other_analyzers
     Send message Distrib_Enable
       and then
     Go to block 2

Block 2:   Wait_for_the_event
     When frame matches LAVc_TR_ReXMT then go to block 3

Block 3:   Send the distributed trigger
     Mark frame
       and then
     Send message Distrib_Trigger
```

## F.13.2 Partner Program

The partner program waits for the distributed enable; then it captures all of the LAN traffic and terminates as a result of either a retransmission or the distributed trigger. Upon termination, this program transmits the distributed trigger to make sure that other LAN analyzers also capture the data at about the same time as when the retransmitted packet was detected on this segment or another segment. After the data capture completes, the data from multiple LAN segments can be reviewed to locate the initial copy of the data that was retransmitted. The partner program is shown in the following example:

```
Store: frames matching LAVc_all
        or Distrib_Enable
        or Distrib_Trigger
       ending with Distrib_Trigger

Log file: not used

Block 1:   Wait_for_distributed_enable
     When frame matches Distrib_Enable then go to block 2

Block 2:   Wait_for_the_event
     When frame matches LAVc_TR_ReXMT then go to block 3

Block 3:   Send the distributed trigger
     Mark frame
       and then
     Send message Distrib_Trigger
```

## F.13.3 Scribe Program

The scribe program waits for the distributed enable and then captures all of the LAN traffic and terminates as a result of the distributed trigger. The scribe program allows a network manager to capture data at about the same time as when the retransmitted packet was detected on another segment. After the data capture has completed, the data from multiple LAN segments can be reviewed to locate the initial copy of the data that was retransmitted. The scribe program is shown in the following example:

```
Store: frames matching LAVc_all
        or Distrib_Enable
        or Distrib_Trigger
       ending with Distrib_Trigger

Log file: not used

Block 1:   Wait_for_distributed_enable
     When frame matches Distrib_Enable then go to block 2

Block 2:   Wait_for_the_event
     When frame matches LAVc_TR_ReXMT then go to block 3

Block 3:   Mark_the_frames
     Mark frame
       and then
     Go to block 2
```

# G

# NISCA Transport Protocol Channel Selection and Congestion Control

## G.1 NISCA Transmit Channel Selection

This appendix describes PEDRIVER running on OpenVMS Version 7.3 (Alpha and VAX) and PEDRIVER running on earliers versions of OpenVMS Alpha and VAX.

### G.1.1 Multiple-Channel Load Distribution on OpenVMS Version 7.3 (Alpha and VAX) or Later

While all available channels with a node can be used to receive datagrams from that node, not all channels are necessarily used to transmit datagrams to that node. The NISCA protocol chooses a set of equally desirable channels to be used for datagram transmission, from the set of all available channels to a remote node. This set of transmit channels is called the **equivalent channel set (ECS)**. Datagram transmissions are distributed in round-robin fashion across all the ECS members, thus maximizing internode cluster communications throughput.

#### G.1.1.1 Equivalent Channel Set Selection

When multiple node-to-node channels are available, the OpenVMS Cluster software bases the choice of which set of channels to use on the following criteria, which are evaluated in strict precedence order:

1. Packet loss history

   Channels that have recently been losing LAN packets at a high rate are termed **lossy** and will be excluded from consideration. Channels that have an acceptable loss history are termed **tight** and will be further considered for use.

2. Capacity

   Next, capacity criteria for the current set of tight channels are evaluated. The capacity criteria are:

   a. Priority

      Management priority values can be assigned both to individual channels and to local LAN devices. A channel's priority value is the sum of these management-assigned priority values. Only tight channels with a priority value equal to, or one less than, the highest priority value of any tight channel will be further considered for use.

   b. Packet size

      Tight, equivalent-priority channels whose maximum usable packet size is equivalent to that of the largest maximum usable packet size of any tight equivalent-priority channel will be further considered for use.

A channel that satisfies all of these capacity criteria is classified as a **peer**. A channel that is deficient with respect to any capacity criteria is classified as **inferior**. A channel that exceeds one or more of the current capacity criteria, and meets the other capacity criteria is classified as **superior**.

Note that detection of a superior channel will immediately result in recalculation of the capacity criteria for membership. This recalculation will result in the superior channel's capacity criteria becoming the ECS's capacity criteria, against which all tight channels will be evaluated.

Similarly, if the last peer channel becomes unavailable or lossy, the capacity criteria for ECS membership will be recalculated. This will likely result in previously inferior channels becoming classified as peers.

Channels whose capacity values have not been evaluated against the current ECS membership capacity criteria will sometimes be classified as **ungraded**. Since they cannot affect the current ECS membership criteria, lossy channels are marked as ungraded as a computational expedient when a complete recalculation of ECS membership is being performed.

3. Delay

   Channels that meet the preceding ECS membership criteria will be used if their average round-trip delays are closely matched to that of the fastest such channel—that is, they are **fast**. A channel that does not meet the ECS membership delay criteria is considered **slow**.

   The delay of each channel currently in the ECS is measured using cluster communications traffic sent using that channel. If a channel has not been used to send a datagram for a few seconds, its delay will be measured using a round-trip handshake. Thus, a lossy or slow channel will be measured at intervals of a few seconds to determine whether its delay, or datagram loss rate, has improved enough so that it meets the ECS membership criteria.

Using the terminology introduced in this section, the ECS members are the current set of tight, peer, and fast channels.

#### G.1.1.2 Local and Remote LAN Adapter Load Distribution

Once the ECS member channels are selected, they are ordered using an algorithm that attempts to arrange them so as to use all local adapters for packet transmissions before returning to reuse a local adapter. Also, the ordering algorithm attempts to do the same with all remote LAN adapters. Once the order is established, it is used round robin for packet transmissions.

With these algorithms, PEDRIVER will make a best effort at utilizing multiple LAN adapters on a server node that communicates continuously with a client that also has multiple LAN adapters, as well as with a number of clients. In a two-node cluster, PEDRIVER will actively attempt to use all available LAN adapters that have usable LAN paths to the other node's LAN adapters, and that have comparable capacity values. Thus, additional adapters provide both higher availability and alternative paths that can be used to avoid network congestion.

### G.1.2 Preferred Channel (OpenVMS Version 7.2 and Earlier)

This section describes the transmit-channel selection algorithm used by OpenVMS VAX and Alpha prior to OpenVMS Version 7.3.

All available channels with a node can be used to receive datagrams from that node. PEDRIVER chooses a single channel on which to transmit datagrams, from the set of available channels to a remote node.

The driver software chooses a transmission channel to each remote node. A selection algorithm for the transmission channel makes a best effor to ensure that messages are sent in the order they are expected to be received. Sending the messages in this way also maintains compatibility with previous versions of the operating system. The currently selected transmission channel is called the **preferred channel**.

At any point in time, the TR level of the NISCA protocol can modify its choice of a preferred channel based on the following:

– Minimum measured incoming delay

  The NISCA protocol routinely measures HELLO message delays and uses these measurements to pick the most lightly loaded channel on which to send messages.

– Maximum datagram size

  PEDRIVER favors channels with large datagram sizes. For example, an FDDI-to-FDDI channel is favored over an FDDI-to-Ethernet channel or an Ethernet-to-Ethernet channel. If your configuration uses FDDI to Ethernet bridges, the PPC level of the NISCA protocol segments messages into the smaller Ethernet datagram sizes before transmitting them.

PEDRIVER continually uses received HELLO messages to compute the incoming network delay value for each channel. Thus each channel's incoming delay is recalculated at intervals of ~2 to ~3 seconds. PEDRIVER then assumes that the network utilizes a broadcast medium (eg. An Ethernet wire, or an FDDI ring). Thus incoming and outgoing delays are symmetrical.

PEDRIVER switches the preferred channel based on observed network delays or network component failures. Switching to a new transmission channel sometimes causes messages to be received out of the desired order. PEDRIVER uses a receive resequencing cache to reorder these messages instead of discarding them, which eliminates unnecessary retransmissions.

With these algorithms, PEDRIVER has a greater chance of utilizing multiple adapters on a server node that communicates continuously with a number of clients. In a two-node cluster, PEDRIVER will actively use at most two LAN adapters: one to transmit and one to receive. Additional adapters provide both higher availability and alternative paths that can be used to avoid network congestion. As more nodes are added to the cluster, PEDRIVER is more likely to use the additional adapters.

## G.2 NISCA Congestion Control

Network congestion occurs as the result of complex interactions of workload distribution and network topology, including the speed and buffer capacity of individual hardware components.

Network congestion can have a negative impact on cluster performance in several ways:

- Moderate levels of congestion can lead to increased queue lengths in network components (such as adapters and bridges) that in turn can lead to increased latency and slower response.
- Higher levels of congestion can result in the discarding of packets because of queue overflow.

- Packet loss can lead to packet retransmissions and, potentially, even more congestion. In extreme cases, packet loss can result in the loss of OpenVMS Cluster connections.

  At the cluster level, these congestion effects will appear as delays in cluster communications (e.g. delays of lock transactions, served I/Os, ICC messages, etc.). The user visable effects of network congestion can be application response sluggishness, or loss of throughput.

Thus, although a particular network component or protocol cannot guarantee the absence of congestion, the NISCA transport protocol implemented in PEDRIVER incorporates several mechanisms to mitigate the effects of congestion on OpenVMS Cluster traffic and to avoid having cluster traffic exacerbate congestion when it occurs. These mechanisms affect the retransmission of packets carrying user data and the multicast HELLO datagrams used to maintain connectivity.

## G.2.1 Congestion Caused by Retransmission

Associated with each virtual circuit from a given node is a transmission window size, which indicates the number of packets that can be outstanding to the remote node (for example, the number of packets that can be sent to the node at the other end of the virtual circuit before receiving an acknowledgment [ACK]).

If the window size is 8 for a particular virtual circuit, then the sender can transmit up to 8 packets in a row but, before sending the ninth, must wait until receiving an ACK indicating that at least the first of the 8 has arrived.

If an ACK is not received, a timeout occurs, the packet is assumed lost, and must be retransmitted. If another timeout occurs for a retransmitted packet, the timeout interval is significantly increased and the packet is retransmitted again. After a large number of consecutive retransmissions of the same packet has occured, the virtual circuit will be closed.

### G.2.1.1 OpenVMS VAX Version 6.0 or OpenVMS AXP Version 1.5, or Later

This section pertains to PEDRIVER running on OpenVMS VAX Version 6.0 or OpenVMS AXP Version 1.5, or later.

The retransmission mechanism is an adaptation of the algorithms developed for the Internet TCP protocol by Van Jacobson and improves on the old mechanism by making both the window size and the retransmission timeout interval adapt to network conditions.

- When a timeout occurs because of a lost packet, the window size is decreased immediately to reduce the load on the network. The window size is allowed to grow only after congestion subsides. More specifically, when a packet loss occurs, the window size is decreased to 1 and remains there, allowing the transmitter to send only one packet at a time until all the original outstanding packets have been acknowledged.

  After this occurs, the window is allowed to grow quickly until it reaches half its previous size. Once reaching the halfway point, the window size is allowed to increase relatively slowly to take advantage of available network capacity until it reaches a maximum value determined by the configuration variables (for example, a minumum of the number of adapter buffers and the remote node's resequencing cache).

- The retransmission timeout interval is set based on measurements of actual round-trip times, and the average varience from this average, for packets that are transmitted over the virtual circuit. This allows PEDRIVER to be more responsive to packet loss in most networks but avoids premature timeouts for networks in which the actual round-trip delay is consistently long. The algorithm can accomodate average delays of up to a few seconds.

#### G.2.1.2 VMS Version 5.5 or Earlier

This section pertains to PEDRIVER running on VMS Version 5.5 or earlier.

- The window size is relatively static—usually 8, 16 or 31 and the retransmission policy assumes that all outstanding packets are lost and thus retransmits them all at once. Retransmission of an entire window of packets under congestion conditions tends to exacerbate the condition significantly.
- The timeout interval for determining that a packet is lost is fixed (3 seconds). This means that the loss of a single packet can interrupt communication between cluster nodes for as long as 3 seconds.

### G.2.2 HELLO Multicast Datagrams

PEDRIVER periodically multicasts a HELLO datagram over each network adapter attached to the node. The HELLO datagram serves two purposes:

- It informs other nodes of the existence of the sender so that they can form channels and virtual circuits.
- It helps to keep communications open once they are established.

HELLO datagram congestion and loss of HELLO datagrams can prevent connections from forming or cause connections to be lost. Table G–1 describes conditions causing HELLO datagram congestion and how PEDRIVER helps avoid the problems. The result is a substantial decrease in the probability of HELLO datagram synchronization and thus a decrease in HELLO datagram congestion.

**Table G–1 Conditions that Create HELLO Datagram Congestion**

| Conditions that cause congestion | How PEDRIVER avoids congestion |
|---|---|
| If all nodes receiving a HELLO datagram from a new node responded immediately, the receiving network adapter on the new node could be overrun with HELLO datagrams and be forced to drop some, resulting in connections not being formed. This is especially likely in large clusters. | To avoid this problem on nodes running:<br>• On VMS Version 5.5–2 or earlier, nodes that receive HELLO datagrams delay for a random time interval of up to 1 second before responding.<br>• On OpenVMS VAX Version 6.0 or later, or OpenVMS AXP Version 1.5 or later, this random delay is a maximum of 2 seconds to support large OpenVMS Cluster systems. |

(continued on next page)

**Table G–1 (Cont.) Conditions that Create HELLO Datagram Congestion**

| Conditions that cause congestion | How PEDRIVER avoids congestion |
|---|---|
| If a large number of nodes in a network became synchronized and transmitted their HELLO datagrams at or near the same time, receiving nodes could drop some datagrams and time out channels. | On nodes running VMS Version 5.5–2 or earlier, PEDRIVER multicasts HELLO datagrams over each adapter every 3 seconds, making HELLO datagram congestion more likely.<br>On nodes running OpenVMS VAX Version 6.0 or later, or OpenVMS AXP Version 1.5 or later, PEDRIVER prevents this form of HELLO datagram congestion by distributing its HELLO datagram multicasts randomly over time. A HELLO datagram is still multicast over each adapter approximately every 3 seconds but not over all adapters at once. Instead, if a node has multiple network adapters, PEDRIVER attempts to distribute its HELLO datagram multicasts so that it sends a HELLO datagram over some of its adapters during each second of the 3-second interval.<br>In addition, rather than multicasting precisely every 3 seconds, PEDRIVER varies the time between HELLO datagram multicasts between approximately 1.6 to 3 seconds, changing the average from 3 seconds to approximately 2.3 seconds. |

# Index

## A



## B

## C

## D

## E

## F



## G



## H

## I



## J



## L

## M

## N

## O



## P

## Q

## R



## S

## T

## U



## V